THÈSE PRÉSENTÉE

POUR OBTENIR LE GRADE DE

DOCTEUR
DE L'UNIVERSITÉ DE BORDEAUX

ECOLE DOCTORALE MATHÉMATIQUES ET INFORMATIQUE

INFORMATIQUE

Par **Gautier Hamon**

Vers des dynamiques ouvertes en vie artificielle et intelligence artificielle: une perspective eco-evo-devo.

Towards open-ended dynamics in Artificial Life and Artificial Intelligence: an eco-evo-devo perspective.

Embodied simulations with rich agent-environment interactions and adaptation at multiple scales.

Sous la direction de : **Clément Moulin-Frier**

Soumis le 02/02/2025, Soutenue le 17/03/2025

Membres du jury :

Pr. Nicolas Bredeche	Professor	Sorbonne Université	Rapporteur
Pr. Daniel Polani	Professor	University of Hertfordshire	Rapporteur
Dr. Nicolas Rougier	Research Director	Inria / Institute of Neurodegenerative Diseases	Président
Dr. Lisa Soros	Roman Family Teaching & Research Fellow	Barnard College	Examinatrice
Dr. Antoine Cully	Associate Professor	Imperial College London	Examinateur
Dr. Clément Moulin-Frier	Researcher	Inria	Director

Vers des dynamiques ouvertes en vie artificielle et intelligence artificielle: une perspective eco-evo-devo

French Abstract

L'évolution naturelle a, au fil de milliards d'années, généré progressivement l'impressionnante diversité de formes de vie complexes qui peuplent notre planète. Ce phénomène illustre ce que nous appelons un processus ouvert (open-ended): un système capable de générer continuellement des structures de plus en plus diversifiées et complexes. Inspiré par ce phénomène ainsi que par d'autres processus ouverts tels que l'apprentissage développemental humain et l'évolution culturelle, cette thèse explore les mécanismes clés qui supportent les processus ouverts et la complexité émergente. Située à l'intersection de la vie artificielle, de l'apprentissage automatique et de l'open-endedness, cette thèse explore, à travers des simulations, la complexité émergente à différents niveaux d'abstraction. Nous mettons l'accent sur l'importance de la dynamique de l'environnement et de son interaction avec les agents adaptatifs dans cette quête de dynamiques ouvertes. En particulier, nous mettons en lumière les effets majeurs des boucles de rétroaction dynamiques, telles que la co-adaptation au sein d'un groupe d'agents ou la causalité réciproque agent-environnement – dans laquelle les agents s'adaptent à l'environnement tout en le modifiant par leur comportement, ce qui, à son tour, modifie l'environnement et façonne leur adaptation. Pour ce faire, nous nous appuyons sur des méthodes de pointe issues de la vie artificielle et de l'apprentissage automatique, notamment les automates cellulaires, la recherche de diversité, la neuroévolution, les systèmes multi-agents et le méta apprentissage par renforcement.

La thèse explore la complexité émergente à différents niveaux d'abstraction. Tout d'abord, elle examine la genèse de l'individualité dans un environnement simulé initialement sans vie, composé d'éléments atomiques simples et de règles physiques locales, explorant aussi comment de tels environnements peuvent amorcer des dynamiques évolutives. Ensuite, en supposant l'existence d'agents et de processus évolutifs, l'accent est mis sur la manière dont les agents adaptatifs modifient activement leurs environnements – potentiellement à leur avantage – modifiant ainsi les pressions évolutives. Ces nouvelles pressions influencent à leur tour les adaptations ultérieures des agents et donc leurs actions sur l'environnement, créant des boucles de rétroaction qui entraînent perpétuellement de nouvelles adaptations de manière potentiellement ouverte. Enfin, la recherche explore comment ces changements environnementaux continus peuvent favoriser le développement de mécanismes d'adaptation plus rapides, permettant aux agents de faire face à cette grande variabilité environnementale. Plus précisément, nous examinons comment des environnements variables peuvent faciliter l'émergence de comportements exploratoires efficaces au sein de groupes d'agents.

En investiguant ces phénomènes, cette recherche apporte des éléments pour concevoir des systèmes capables de démarrer et de maintenir des processus ouverts, reflétant ainsi la richesse et la complexité adaptative du monde naturel — de l'origine de la vie à l'évolution d'agents généralistes.

Mots clés

Open-endedness; Systèmes complexes; Vie artificielle; Algorithmes évolutionnaires; Apprentissage par renforcement multi-agent; Meta-apprentissage.

Towards open-ended dynamics in Artificial Life and Artificial Intelligence: an eco-evo-devo perspective

Abstract

Natural evolution has, over billions of years, gradually generated the astonishing diversity of complex life forms that populates our planet. This phenomenon exemplifies what we call an open-ended process: a system capable of continuously generating increasingly diverse and complex structures. Inspired by this phenomena as well as other open-ended processes such as human developmental learning and cultural evolution, this thesis investigates key mechanisms that underpin open-ended processes and emergent complexity. Situated at the intersection of artificial life, machine learning, and open-endedness, this thesis explores, in simulations, emergent complexity across varying levels of abstraction. We focus on the importance of environment dynamics and its interplay with adapting agents in this quest of open-ended dynamics *in silico*. In particular, we highlight the major effects of feedback loop dynamics, such as co-adaptation in a group of agents or agent-environment reciprocal causation – wherein agents adapt to the environment but also alter it through their own behavior, which in turn modify the environment and shape their adaptation. For this aim, we rely on diverse state-of-art methods from artificial life and machine learning, including cellular automata, diversity search, neuroevolution, multi-agent systems and meta reinforcement learning.

The thesis explores emergent complexity at different levels of abstraction. First, it explores the genesis of individuality within an originally lifeless simulated environment composed of simple atomic elements and local physical rules, also probing how such environments can bootstrap evolutionary dynamics. Next, assuming the existence of agents and evolutionary processes, the focus shifts to how adapting agents actively modify their environments – potentially to their advantage –thereby altering evolutionary pressures. These new pressures, in turn, influence the agents' subsequent adaptations and therefore actions on the environment, creating feedback loops that perpetually drive new adaptations in a potentially open-ended way. Finally, the research explores how these continual environmental changes may foster the development of faster adaptation mechanisms, enabling agents to cope with this high environmental variability. Specifically, we examine how variable environments can facilitate the emergence of efficient exploratory behaviors within groups of agents.

By investigating these phenomena, this research contributes foundational insights toward designing systems capable of bootstrapping and sustaining open-ended processes, ultimately reflecting the rich, adaptive complexity of the natural world – from the origins of life to the evolution of generalist agents.

Keywords

Open-endedness; Complex systems; Artificial life; Evolutionary algorithms; Multi-agent reinforcement learning ; Meta-learning.

Inria FLOWERS team

Extended French Abstract

L'évolution naturelle a, au cours de milliards d'années, transformé de simples organismes unicellulaires en l'incroyable diversité de formes de vie complexes que nous observons aujourd'hui. Au cours d'un seul essai, ce merveilleux processus a continuellement accru la complexité et la diversité des organismes qu'il a créés, apparemment sans limite. Ce phénomène illustre ce que nous appelons un processus ouvert — un système capable de générer continuellement des structures de plus en plus diversifiées et complexes. Inspiré par ce phénomène ainsi que par d'autres processus ouverts tels que l'apprentissage développemental humain et l'évolution culturelle, ce travail explore les mécanismes clés qui sous-tendent les processus ouverts et la complexité émergente. Située à l'intersection de la vie artificielle, de l'apprentissage automatique et de l'open-endedness, cette thèse explore, à travers des simulations, la complexité émergente à différents niveaux d'abstraction. Nous mettons l'accent sur l'importance de la dynamique de l'environnement et de son interaction avec les agents adaptatifs dans cette quête de dynamiques ouvertes. En particulier, nous mettons en lumière les effets majeurs des boucles de rétroaction dynamiques, telles que la co-adaptation au sein d'un groupe d'agents ou la causalité réciproque agent-environnement — dans laquelle les agents s'adaptent à l'environnement tout en le modifiant par leur comportement, ce qui, à son tour, modifie l'environnement et façonne leur adaptation. Pour ce faire, nous nous appuyons sur des méthodes de pointe issues de la vie artificielle et de l'apprentissage automatique, notamment les automates cellulaires, la recherche de diversité, la neuroévolution, les systèmes multi-agents et le méta-apprentissage par renforcement.

Le premier chapitre 0, introduit les concepts nécessaires à la compréhension de la thèse et présente la structure globale de la thèse. Il commence par introduire le concept d'open-endedness et son importance. Il présente ensuite le domaine de la vie artificielle et son approche se rapprochant d'une recherche de complexité émergente (une approche axée sur la complexité émergente). Cela se différencie de l'approche classique moderne en apprentissage automatique qui optimise des architectures cognitives prédéfinies pour des objectifs fixes méticuleusement construits (une approche axée sur les objectifs). Cette thèse se place dans la continuité de travaux récents qui proposent d'abandonner cette approche axée sur les objectifs pour se concentrer sur des algorithmes capables de générer des structures plus complexes, si possible de manière ouverte. En particulier, cette thèse se place à la frontière entre approches en vie artificielle et approches en apprentissage automatique classique. Notamment, cette thèse se concentre sur l'adaptation à multiples échelles d'agents ainsi que leurs interactions avec des environnements aux dynamiques complexes. En particulier, elle met en lumière les boucles de rétroaction agent-agent et agent-environnement et comment celles-ci peuvent mener à des augmentations de la diversité et complexité notamment vers l'émergence d'agents plus généralistes.

Le premier chapitre se poursuit en introduisant les différentes boucles d'adaptation dans le monde naturel et leur équivalent en simulation: l'auto-organisation et la maintenance autonome, l'évolution, l'apprentissage développemental, l'évolution culturelle, et comment celles-ci interagissent entre elles.

Nous abordons ensuite l'importance de l'environnement, et en particulier la causalité réciproque entre adaptation des agents et environnement, pour obtenir des dynamiques ouvertes. En effet, les agents ne sont pas seulement les produits de l'évolution, de par leurs actions sur l'environnement (construction de niche) ils sont des acteurs de l'évolution. Lorsque les agents amènent des changements dans l'environnement, ceux-ci vont changer les opportunités et pressions de l'environnement menant à de nouvelles adaptations de la part des agents. Ces nouvelles adaptations peuvent alors de nouveau mener à des changements d'environnement, etc. Ces causalités réciproques entre agent et environnement sont appelées dynamiques éco-évolutionnaires et peuvent potentiellement mener à des complexifications ouvertes. Ce chapitre présente également les dynamiques entre plusieurs agents et comment celles-ci peuvent aussi mener à une augmentation de la complexité et de la diversité par exemple à travers la compétition.

Le premier chapitre se finit par la présentation de la structure de la thèse présentée ci-dessous:

Dans le chapitre 1, nous considérons un environnement dépourvu de vie, dans un état initial où il n'existe littéralement *aucun corps* (et donc aucune perception, aucune action, aucun agent, aucune

évolution). Notre objectif est d'étudier **comment certaines parties d'un tel environnement inanimé pourraient s'auto-organiser en structures donnant lieu à des proto-formes de vie fonctionnelles et amorcer leur évolution**. Pour ce faire, nous nous appuyons sur des automates cellulaires continus récents, en utilisant des algorithmes de recherche par diversité pour explorer leur espace des paramètres à la recherche de phénomènes d'auto-organisation pertinents.

Dans une première contribution, nous appliquons des algorithmes de recherche par diversité et d'apprentissage par curriculum à des automates cellulaires continus afin de rechercher des règles du système conduisant à l'émergence systématique de structures auto-organisées affichant des capacités sensori-motrices de base – c'est-à-dire des proto-agents capables de réagir à des perturbations de l'environnement. De manière intéressante, nous découvrons des structures auto-organisées qui semblent capables de prendre des décisions à l'échelle macroscopique uniquement grâce à la dynamique collective de nombreuses parties atomiques, c'est-à-dire sans aucune notion de "cerveau" central, de capteurs ou d'actionneurs. De plus, ces agents auto-organisés montrent d'impressionnantes capacités de généralisation à des conditions non observées lors de la recherche.

Notre deuxième contribution explore l'auto-organisation de dynamiques évolutives dans un environnement similaire dépourvu de vie. En particulier, nous étendons les automates cellulaires continus utilisés dans la première contribution, ce qui nous permet d'introduire des simulations "multi-espèces", où des structures auto-organisées régies par des règles de mise à jour différentes peuvent coexister. Dans ces simulations multi-espèces, nous observons des dynamiques évolutives émergentes découlant de la physique du système, sans recours à un algorithme évolutif externe. En particulier, nous observons une activité évolutive émergente résultant des interactions coopératives ou compétitives entre diverses "espèces" auto-organisées, encore une fois uniquement grâce à la dynamique collective de nombreuses parties atomiques.

Le chapitre 1 démontre ainsi l'émergence d'individualité, de cognition de base (sous forme de capacités sensori-motrices) et d'évolution dans un environnement initialement dépourvu de vie, par le biais d'interactions entre des éléments atomiques simples. Cela aboutit finalement à des environnements avec des **agents adaptatifs, bien séparés de l'environnement**, qui prolifèrent et meurent via des interactions avec des entités voisines. Dans la suite, nous considérons ensuite une dichotomie plus classique entre un environnement et des agents interagissant avec celui-ci, prééquipés de capteurs, d'actionneurs et de capacités de prise de décision.

Dans le chapitre 2, nous étudions l'interaction entre l'adaptation des agents et la dynamique environnementale avec des agents incarnés bien séparés de l'environnement. Plus précisément, nous explorons les dynamiques éco-évolutives émergentes et les phénomènes de construction de niche, en nous posant la question suivante : **Comment des populations d'agents adaptatifs éco-conçoivent-elles leur propre environnement en présence de rétroactions éco-évolutives ?**. Nous nous concentrons sur deux contributions principales.

La première contribution présente un système où les agents évoluent continuellement sans aucune réinitialisation de l'environnement ou de la population, permettant des rétroactions éco-évolutives. L'environnement est un vaste monde en grille avec une génération complexe de ressources spatio-temporelles, contenant de nombreux agents, chacun étant contrôlé par un réseau neuronal récurrent évolutif et se reproduisant localement en fonction de leur physiologie interne. Nous montrons que la neuroévolution peut fonctionner dans un cadre multi-agents non épisodique écologiquement valide, trouvant des stratégies collectives de collecte durable en présence d'une interaction complexe entre dynamiques écologiques et évolutives.

La seconde contribution explore l'émergence de pratiques agricoles au sein d'une population d'agents utilisant l'apprentissage par renforcement. Situés dans un environnement avec différentes ressources en compétition, ces agents apprennent à "éco-engineer" leur environnement pour promouvoir la prolifération de ressources bénéfiques. Cette convergence vers des stratégies collectives de construction de niche met en évidence leur capacité à modifier leur environnement à leur avantage.

Le chapitre 2 introduit ainsi la construction de niches et les rétroactions éco-évolutives. En particulier, cette interaction complexe entre agents adaptatifs et environnement peut conduire à des environnements avec des variations rapides auxquelles les agents doivent faire face.

Dans le chapitre 3, nous contrôlons la variabilité environnementale et étudions **comment des stratégies d'exploration avancées, génériques, collectives et potentiellement ouvertes peuvent émerger chez des agents adaptatifs exposés à une forte variabilité environnementale**.

Notre première contribution exploite des tâches hiérarchiques générées procéduralement pour étudier l'émergence de l'exploration collective dans un groupe d'agents indépendants. À partir de l'entraînement sur une distribution diversifiée de tâches où les règles sous-jacentes doivent être découvertes, les agents apprennent à explorer collectivement les affordances de l'environnement. Ils montrent également une généralisation intéressante à de nouvelles tâches et à des chaînes de tâches plus longues (avec plus d'objets, etc.) non observées pendant l'entraînement.

Dans la seconde contribution, nous passons à des groupes d'agents indépendants avec un mécanisme d'exploration "autotelic" (créant leurs propres objectifs) prédéfini et étudions l'émergence d'une intentionnalité partagée pour faire face à la variabilité induite par d'autres agents explorateurs. En particulier, à partir de la maximisation indépendante de leurs récompenses, les agents apprennent à communiquer et à aligner leurs objectifs, atteignant finalement un apprentissage plus efficace par rapport à des agents choisissant leurs objectifs indépendamment.

En investiguant ces phénomènes, cette recherche apporte des éléments pour concevoir des systèmes capables de démarrer et de maintenir des processus ouverts, reflétant ainsi la richesse et la complexité adaptative du monde naturel — de l'origine de la vie à l'évolution d'agents généralistes.

La thèse se conclut avec des ouvertures sur des perspectives sur la réunion des éléments présentés dans chaque chapitre en une seule simulation aux dynamiques intéressantes. En particulier, nous discutons des éléments que nous pensons intéressants pour construire un environnement avec des dynamiques ouvertes qui pourraient permettre de mener à une diversité d'agents complexes et aux capacités généralistes. Nous présentons aussi des perspectives sur la dynamique des agents, ainsi que sur les interactions multi-agents. Nous finissons par des discussions générales sur l'équilibre entre biais et complexité émergente dans les simulations open-ended, ainsi que sur les challenges inhérents à la mesure de dynamiques ouvertes, et la potentielle utilité des simulations présentées pour mieux comprendre l'évolution et en particulier l'évolution humaine.

Disclaimer : Large language models have been used solely as a reformulation tool throughout this thesis, only to improve the clarity and flow of the text sporadically.

Contents

Introduction 0

0.1 Open-endedness

What is it ? Natural evolution has, over billions of years, transformed simple single-celled organisms into the astonishing diversity of complex life forms we see today. This remarkable process increased the complexity of organisms it created over and over, seemingly without limit. This phenomenon exemplifies what we call an open-ended process—a system capable of continuously generating increasingly diverse and complex structures.

Natural evolution is not the only example of such a process. It has also given rise to other open-ended phenomena. For instance, human skill acquisition can be considered open-ended; humans are capable of continually learning and mastering increasingly complex tasks throughout their lifetimes, often building on existing knowledge like advanced motor skills or intellectual capabilities. Similarly, cultural evolution exhibits open-ended dynamics. The cumulative development of culture—manifested in fields such as mathematics, technology, and art—has led to increasingly sophisticated and abstract achievements over generations.

Why is it important ? Understanding the principles underlying open-endedness is an important endeavor for the scientific community [1]. In Life Science, the challenge is to better understand the mechanisms having generated the immense diversity and complexity of life forms on Earth. In Computer Science, implementing a process capable of generating a diversity of increasingly complex problems and solutions would have tremendous implications across a wide range of scientific domains. Harnessing the power of open-ended processes could transform domains such as drug discovery, protein search, engineering problems, art, and even science itself [2]. For example, being capable of reproducing a process similar in its dynamics to open-ended natural evolution could potentially allow us to achieve human-like generally intelligent artificial agents and potentially even more [3].

Artificial Life and the "bottom up" approach The field of Artificial Life (Alife) has itself extensively focused on simulating artificial ecosystems and open-ended evolution, with the objective of implementing a process able to spontaneously generate increasingly diverse and complex structures, emerging from the dynamics of the system itself (hence the term "bottom-up") [4].

In particular, Alife works often search for the necessary conditions for complexity to emerge and flourish [5, 6]. They often focus on implementing environments with simple but rich dynamics that can lead to the emergence of higher complexity. For instance, self-organizing

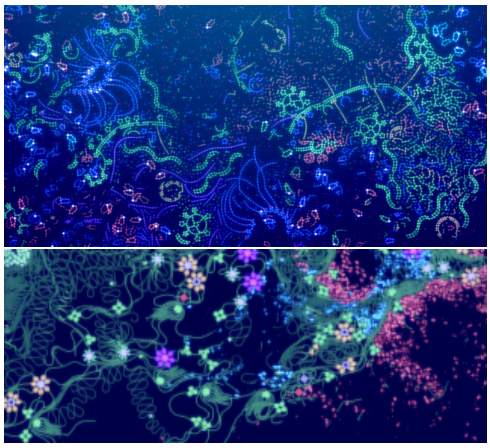


Figure 1: **Artificial life**. Winner of the Alife creature competition 2024. Alien project *https://alien-project.org/index.html*

systems – systems composed of simple atomic entities interacting together through simple rules potentially leading to complex macro intelligent entities – are often used as substrates for Alife works [7–9].

This "bottom-up approach" is closely related to the idea of open-ended processes which start from simple conditions and increase in complexity over and over. In fact, implementing such evolution and minimal environment might be much easier than directly engineering complex intelligent artificial agents and might also lead to unexpected interesting behaviors [10].

However, while we have examples of open-ended processes in the natural world to get inspiration from, we are still far from being able to reproduce truly open-ended processes through simulations or to use them to achieve "generally" intelligent artificial agents. Surprisingly, while natural evolution is our best example of a process capable of generating such interesting intelligent agents, current Artificial Intelligence (AI) techniques have little in common with the way natural evolution works.

Machine learning is currently mostly top down. In fact, even evolutionary algorithms which are historically inspired by natural evolution (see Sec.0.2.2) most often adopt a "top down" approach: their main application is to optimize a solution to a target problem [11, 12] (with notable exceptions such as Quality-Diversity algorithms that we discuss below).

Current state of the art machine learning (ML) techniques in general tend to focus on this optimization of a highly structured engineered cognitive architecture towards a single predefined fixed objective. Improvements then come from meticulously engineering better cognitive architecture or objective functions [13]. In particular, with the advent of neural networks as general function approximations, a lot of effort has been put into trying to improve their architecture: from multi layers perceptron [14], to convolutional neural networks (CNNs)[15], recurrent neural networks (RNNs) [16–18], transformers [19], and more recently, structured state-space models (SSMs) [20]. Another focus of current machine learning is the data used to instill knowledge in these cognitive architectures, especially with the use of supervised learning – training on a labeled training set of input-output pairs – as the main tool to obtain "intelligent" artificial agents.

While these techniques achieved impressive performances for specific problems, for example in computer vision [21, 22] or recently in natural language processing [19, 23, 24], their generalization capabilities and reasoning abilities are still questioned [25–28]. In addition, the emergence of capabilities beyond the training distribution is still up to debate [26, 29], potentially hinting toward additional components needed to get beyond the training data or even the need for a complete switch of paradigm. In fact, while scaling the amount of data or cognitive architecture size has been shown to be effective —often referred to as the "bitter lesson" of machine learning [30]—, its sufficiency to achieve truly general intelligent artificial agents is

still an open question, in particular in the quest for artificial agents capable of continual improvement.

Change of perspective: from "performance driven approaches" to "emergent complexity approaches". Achieving open-endedness might require a change of perspective compared to the dominant approach in machine learning. This dominant approach relies on meticulously engineered cognitive architectures, optimized against pre-defined objective functions, and evaluated through benchmarks capturing relevant features of intelligence. We propose to call it "performance-driven approaches", in the sense that proposed architectures are optimized according to a user-defined metric, and argue that it contrasts with open-ended processes such as natural evolution.

"... The major inspiration for both evolutionary computation and genetic programming, natural evolution, innovates through an open-ended process that lacks a final objective." [31]

In fact, it is far from trivial to consider natural evolution in terms of objectives, e.g. maximizing survival and reproduction. For instance, some species have evolved towards a very short life time (e.g. ephemeral species such as flies) and others give birth to very few offsprings (e.g. humans), yet manage to play an important role in their respective ecosystems. Instead, natural evolution seems to be better characterized by the notion of open-endedness than by the notion of objective [10]. Some approaches in ALife, and to a lesser extent in AI, have embraced this view and evaluate their simulations in terms of emergent complexity instead of explicit performance [5, 32–40]. Some authors have theorized this view and proposed to abandon, or at least to reconsider, the "myth of the objective" [10, 31, 41]. We propose to call such approaches *complexity-driven*, in the sense that they consider intelligence as the emergent product of a dynamical system, in which agents continually adapt to ever-changing environmental dynamics (by opposition to the *performance-driven approaches* described above, which considers intelligence as a measurable objective).

In particular, the complexity-driven approach tries to understand the necessary condition for a process leading to emergent complexity and general intelligence from a simpler state [3, 5, 10], rather than trying to directly build it. What we call complexity-driven encompasses both: 1) systems where there is no notion of objective at all, where agents adapt through the dynamic of the system–what is called "environment-driven" in Bredeche and Montanier (2012) [42]; and 2) adaptive systems with implicit objectives (also introduced in [42]) where the objective function only gives partial information on how to act in a task but might lead to emergent complexity, such as optimizing for the maximization of energy which can lead to various complicated behaviors.

While the question remains open and the boundary between implicit and explicit objectives may be blurry, from a practical perspective, searching for the necessary conditions to foster the emergence of complex artificial agents—such as generalist artificial agents—might be significantly easier than attempting to engineer them directly [3]. For instance, in computer vision, the shift from engineered representations to learned representations [15, 21] demonstrated that allowing a system to discover solutions autonomously is often far more

effective than designing those solutions manually. In addition, just as natural evolution flourished by producing a **diversity** of strategies to "solve problems," approaches based on emergent complexity could likewise yield a variety of sophisticated and diverse "solutions" (compared to the performance driven approach often aiming for a single solution).

Convergence between machine learning, open-endedness and artificial life. In fact, recent work suggests that ideas from open-endedness could provide a road-map for overcoming ML's limitations [1, 3, 43]. For example, training systems on an open-ended distribution of tasks has shown promise in developing general skills [44, 45], showing the importance of training agents on a wide diversity of rich environments. A very promising avenue towards this are processes that generate their own problems and try to solve them, in an open loop [46–49]. This kind of work necessitates a component generating the problems and an agent capable of adaptation to solve them.

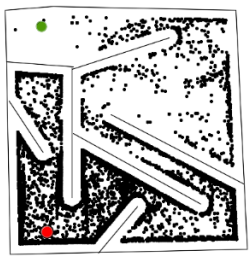
"To achieve open-endedness, a model must not only consume knowledge from pre-collected feedback [...], but also generate new knowledge, in form of hypotheses, insights or creative outputs beyond the human curated training data. A self-improvement loop should allow the agent to actively engage in tasks that push the boundary of its knowledge and capabilities..." [43]

In addition to ML benefiting from the ideas of open-endedness, we observe more and more a kind of convergence between ideas of current ML techniques and ideas inspired by open-endedness and Alife, where both fields benefit from each other.

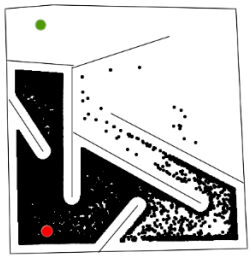
In particular, Alife works also benefit from machine learning, for example with the cognitive architecture developed in classical machine learning being used in Alife studies [34, 50].

Self-organizing systems used as substrate in environments in Alife are also proposed as candidates for cognitive architecture in ML [51–55], for their interesting robustness potentially helping ML with the generalization problem (despite not being state of the art at the moment). They might also serve as a general architecture whose topology (connections, structure) might themselves adapt by self-organization [56, 57], in opposition to fixed architecture in classical ML, potentially helping in the quest of systems continually increasing in complexity.

Another line of work in machine learning takes inspiration from natural evolution and open-endedness and tries to find a diversity of solutions– a field known as quality diversity [58–62]. These works often use machine learning techniques in combination with diversity search in order to build an ideally never-ending archive of solutions to a problem, potentially getting more and more complex using previous solutions as stepping stones for new ones. In particular, the diversity part of these methods might allow to obtain better solutions by avoiding the trap of local optima [61, 63] (Fig.2).



(b) Hard Map Novelty



(d) Hard Map Fitness

Figure 2: Maze task. The goal position is the green circle, and the start position is the red circle. The black dots represents the end position of each trials. Objective based algorithms (d) easily fall into local optima, as exemplified by the maze here where the majority of trials go toward the dead end as it makes the solution go closer to the goal. On the other hand, methods based on novelty try to cover the maze and therefore explore the whole maze ultimately leading to trials reaching the goal. Figure from [61].

Finally, foundation models in machine learning (or other highly engineered components) might be a good starting point to help bootstrap an open-ended loop, for example, with components capable of generating new problems, such as Large Language Models (LLMs)[46, 64, 65]. However, studying the minimal setup to bootstrap an open-ended process might as well be easier (and less costly). In particular, better understanding the process of open-endedness in minimal setups such as simple artificial life ecosystems [5] might be beneficial

to apply it in more complex setups with more complicated cognitive architecture in ML.

Towards open-ended dynamics in Artificial Life and Artificial Intelligence: an eco-evo-devo perspective. This thesis explores the intersection of machine learning, artificial life (ALife), and open-endedness, aiming to bridge the gap between these fields. **The long-term aspiration is to establish the foundational conditions necessary for the open-ended evolution of generalist agents.**

Recognizing that achieving this ambitious goal extends beyond the scope of a single thesis, our focus here is to examine specific transitions and components that we consider critical to simulate open-ended dynamics in artificial systems.

In particular, we will focus on the interaction between adaptive artificial agents and the dynamics of the environment at different scales, exploring how these interactions can lead to emergent complexity. More precisely, here are the main ingredients that we'll explore in this thesis and develop in the rest of this introduction :

- **Adaptation Across Multiple Scales** (Sec.0.2). In the natural world, adaptation operates across multiple spatio-temporal scales: evolutionary, developmental and cultural. These different scales strongly interact with each other (Sec.0.2.5). We will argue below that adaptation at multiple scales is central to open-ended processes and we will rely on state-of-the art AI and ALife methods to implement it in simulations.
- **Dynamic environments** (Sec.0.3.1). The morphological and behavioral complexity of living beings on Earth strongly depends on the the complexity of the environment they live in [66]. This environmental complexity is driven by the multiscale dynamics of environmental changes, such as seasonal cycles, ecosystem dynamics and the presence of other cooperating or competing agents. The contributions of this thesis will place environmental design at the center of our computational approach, proposing diverse simulated environments exposing adaptive agents to constantly changing constraints and opportunities.
- **Feedback loops between agents' adaptation and the dynamics of their environment** (Sec.0.3.2). Based on both previous points, open-ended dynamics is often conceived as an emergent property of complex systems able to self-generate their own problems and solutions. For instance, while the properties of an environment implies selective pressures on evolving organisms, their own evolution in turn modifies environmental properties and their resulting selective pressures. Such feedback loops between agents' adaptation and the dynamics of their own environment is central in most of our contributions.

In the following of this introduction, we will develop the three main points made above from a biological, ecological, and computational perspective. Then we will introduce in more detail the objectives and contributions of the thesis.

0.2 Mechanisms of adaptation at multiple scales

The first component that we'll cover is agent adaptation; the process that allows an agent to cope with environmental complexity. We cover in this section different levels of adaptation: self-maintenance (Sec.0.2.1), evolution (Sec.0.2.2), developmental learning (Sec.0.2.3), cultural evolution (Sec.0.2.4) and finally how they interact altogether (Sec.0.2.5)– covering them in a broad manner in this introduction and referring to the appropriate sections of the contributions for more details and related works. For each of the adaptation levels, we draw parallels between existing frameworks in life science (biology, ecology, and cognitive science) and computer science (artificial life and artificial intelligence).

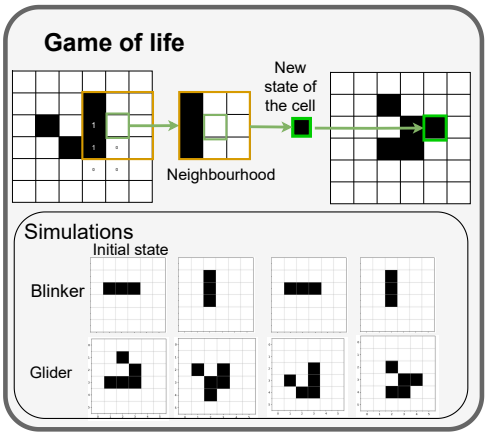


Figure 3: **Conway's Game of Life cellular automata**[68]. Each cell is updated based on its neighbourood. The self-organization of a collective of simple cells can enable the emergence of localized macro structures, such as the "blinker", and sometimes mobile ones like the "glider."

0.2.1 Self-organization and autopoiesis

We here concisely introduce the concepts of autopoiesis and enactivism and refer to Sec.1.2 for a more detailed review.

The most basic level of adaptation is maintaining its integrity despite potential perturbation by the environment, i.e. the ability to self-maintain and self-regulate. A key example of this is homeostasis, the process by which a system maintains a stable internal state, such as balancing pH levels or temperature within a specific range. This capacity for self-maintenance lies at the heart of the concept of autopoiesis, introduced by Maturana and Varela [67]. Autopoiesis refers to a system's ability to produce and sustain itself by generating and preserving its own components. For Maturana and Varela, autopoiesis is not only a defining characteristic of life but also a basis for cognition: any autopoietic system inherently possesses some level of cognitive capacity. Examples of autopoietic systems include living cells—capable of synthesizing proteins, repairing membranes, and maintaining internal balance—or entire organisms like plants and animals, which achieve cellular reproduction, tissue repair, and homeostasis (e.g., regulation of temperature, pH, and energy balance).

1: **Cellular automata (CA)** are, in their classic form, a grid of "cells" that evolve through time via the same local "physics-like" laws: Each cell sequentially updates their state based on the states of their neighbours. See Sec.1.1.1 for more details.

In Artificial life, autopoiesis is a central concept often framed through the lens of self-organization. In those works, agents come to existence through the self-organization of simple parts and maintain their integrity and stability through their coordination (often displaying self-repairing capabilities). Examples of such works span several artificial substrates such as cellular automata (CA) [1][8, 9, 69–71], artificial chemistries and generative grammar [72–74], modular robots [38, 75], and soft robots [76]. In particular, cellular automata, like Conway's Game of Life (Fig.3), have served as simple yet powerful models for studying the principles underlying autopoiesis [77–80].

This perspective on agency aligns closely with the so-called enactivist framework (Fig.4) [81, 82], which considers that the agent must come to existence through the coordination of the low-level elements and that every part of an individual contributes to cognition. In this view,

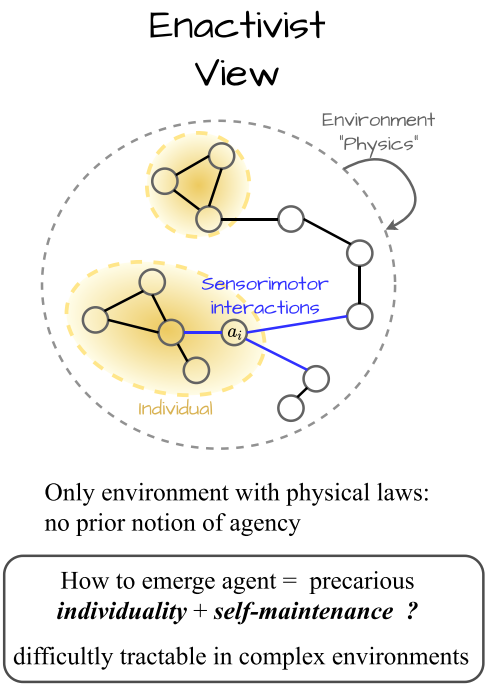
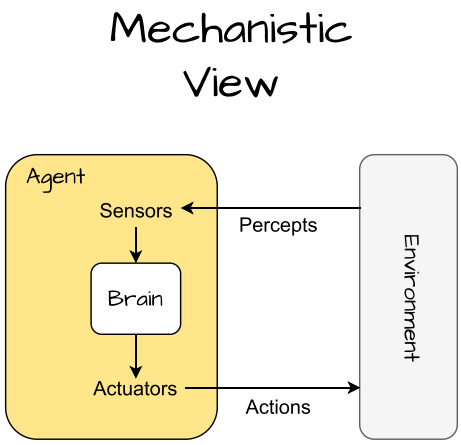


Figure 4: Enactivist framework

cognition is an embodied, distributed process, deeply rooted in the system's material and dynamic organization.

In contrast, classical AI and machine learning typically operate under a "mechanistic framework" (Fig.5). Here, the agent is assumed to have a well-defined physical body and an information-processing brain, interacting with the environment through predefined sensors and actuators. The agent's body is treated as separate from the environment, and its structural integrity is presumed unperturbed by the environment. This approach largely bypasses questions of body constitution, focusing instead on optimizing a centralized control unit for predefined tasks. The enactive view of intelligence—emphasizing self-organization and the emergence of agency from low-level components—remains underexplored in mainstream AI.



One assumes the pre-existence of a body with sensors and actuators, through which an agent can interact with its environment

Figure 5: Mechanistic framework

However, several efforts at the intersection of Alife and classical AI propose to study Artificial Intelligence under the lens of the enactivist framework and self-organization, both advocating for their relevance to better understand the living, but also as a tool to build more robust artificial agents [51, 52]. Examples include cellular automata being used as controllers [53], systems leveraging the self-organization in Hebbian networks [54], and sensorimotor controls in self-assembling robots [75].

Another line of work directly implements homeostasis as a feature of the cognitive architecture [83] or as an objective to optimize for [84].

The concept of autopoietic agents, which "generate their own parts", is also closely related to agents that generate new individuals by creating new parts. In fact, self-replicators are also a major focus of autopoietic systems study. In particular, Alife works also often study the self-organization of self-replicators [6, 71, 72, 85–88] as it is a necessary condition for evolution to emerge in such systems.

0.2.2 Evolution

Biological evolution is perhaps the most typical example of an open-ended process, where organisms adapt their phenotypic traits through the transmission, variation, and selection of genetic material and under environmental selective pressures. The principle is that, depending on the environment, certain traits may be favored, giving survival or reproduction advantages, ultimately allowing those traits to be more transmitted to the next generation. Over time, this process leads to individuals better adapted to their environment. For example, Darwin finches are birds whose beaks are sensibly different depending on the availability of resources in their local niche: those in an area where nuts are more present developed over time a thicker beak while others developed a slender beak to eat nectar or pluck small insects more easily [89, 90]. In a changing environment this type of dynamic can possibly lead to increasingly diverse and complex beings.

As mentioned previously, the field of artificial life –which explores "life as it could be"– has been trying to reproduce natural evolution

in simulated worlds [5, 32–38, 42, 91]. While a large body of works in Alife directly implements well-defined evolutionary mechanisms in the system, the Alife community is also interested in the minimal requirement for evolution itself to emerge from the dynamic of the system. In particular, as a first step toward evolutionary dynamic, a lot of focus has been put on having self-replicators in self-organizing systems [6, 72, 85–88] potentially also displaying increase in complexity [71]. Those systems, however, still fail to achieve truly open-ended dynamics.

Natural evolution has also been an inspiration for optimization algorithms such as evolutionary algorithms (EA) [11, 12] that are also often used as a way to optimize for a certain metric in a "performance driven" manner. Those approaches often require the definition of a "fitness function" that will be used to measure agent performance. The methods span from more or less ecologically plausible but still take inspiration from either population mutation, crossover, or selection. In fact, most EAs share a similar procedure of applying transmission, variation, and selection on a population of potential solutions to an optimization problem. They mostly differ in how they encode the search space and the population : genetic algorithms [92, 93] encode each individual with a genome (often binary), genetic programming [94] is similar to genetic algorithms but uses programs as the search space; evolutionary strategies such as CMA-ES [95] represent individuals with real numbers, and also often encode the population as a probability distribution [96–98].

Neuroevolution is a specific instance of evolutionary algorithms where the search space is artificial neural networks. This approach can explore the weights of a neural network [99] or also its architecture [100, 101]. With the recent advances in computational capabilities, neuroevolution has been shown to be potentially competitive with classical machine learning [99, 102, 103]. We refer to 2.1.2 for more details on Neuroevolution.

0.2.3 Developmental learning

Natural selection has given rise to other adaptation processes operating at smaller timescales. While evolutionary changes unfold across generations, organisms in nature might also possess the remarkable ability to acquire skills during their individual lifetimes through developmental learning. Consider human development: an infant begins with limited capabilities but rapidly develops motor skills, and this learning continues throughout life - going from basic movements to complex abilities like playing musical instruments or computer programming. This developmental trajectory is shaped both by the individual's environment (including social peers) and intrinsic motivation [104–107] shaping what they choose to explore and learn.

In AI, Reinforcement learning (RL) is often used to model developmental learning (at least on some level). RL provides a computational framework that captures aspects of developmental learning through trial and error [108]. In this paradigm, agents learn by receiving positive or negative feedback from their environment (or their internal

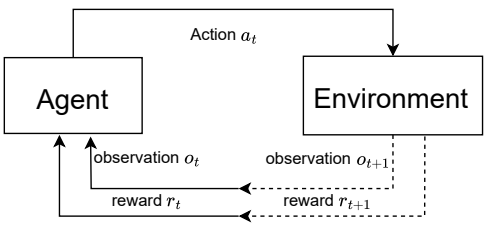


Figure 6: The Reinforcement Learning loop.

model), gradually favoring actions that lead to (future) positive rewards while avoiding those that result in (future) negative rewards. In particular, RL agents, through the interaction with the environment, learn a *policy function* that maps *observations* (sensory inputs) to *actions* (motor outputs) (Fig.6). We refer to Sec.3.2.4 for a formal definition of RL.

The field of reinforcement learning led to impressive results in games (for example in chess or GO [109, 110]), video games (e.g. Atari) [110–112], and even recent video games requiring advanced real time strategy [113–115], as well as robot control [116–121].

Reinforcement learning also showed interesting results in agents being capable of learning a variety of tasks. In particular, the field of goal-conditioned RL which trains a single agent over several tasks, giving task descriptions to the agent as input to the policy [122, 123].

However, implementing truly continuous general learning in artificial agents remains a challenge. RL still faces challenges for continual learning such as plasticity loss [124–130], catastrophic forgetting [131, 132], and most importantly exploration problems [133]. To enhance exploration capabilities of agents, methods relying on **intrinsic motivation** strategies have been introduced [134, 135]. Those methods give higher incentives to explore, for example, giving a reward for novel state [136, 137], model prediction error [138–140] ,surprise [141], (in)competence [142], or empowerement (how much the agent can "change its environment") [143].

In addition, RL as it is described most of the time already assumes that the task is set by the user and given to the agent. This omits an important part of the developmental learning process in humans where the selection of a task or a goal is done by the agent itself. In fact, this goal selection by the agent is an important part of the whole developmental learning process as it will shape the learning trajectory of the agent (what he chooses to learn and in which order). Works on autotelic learning [144, 145] (from the Greek auto (self) and telos (goal)) focus on implementing learning agents that select their own goals (and actively try to achieve them) for a more complete picture of developmental learning. The implemented goal selection processes also often include aspects of curiosity, taking inspiration from human cognition. We refer to Sec.3.2.3 for more details on autotelic learning.

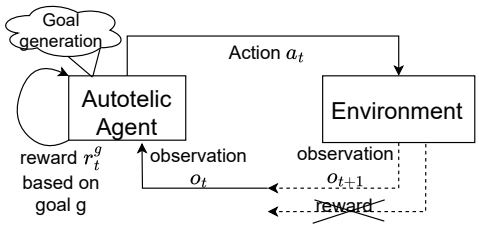


Figure 7: **Autotelic agent**: agents that generate their own goal and actively learn how to achieve them.

0.2.4 Cultural evolution

In the human species, and potentially others, skills and knowledge can be socially transmitted between individuals, potentially adapting them to their own incentives. This process can result in the cumulative accumulation of diverse and complex cultural traits: an open-ended process called Cultural Evolution [146]. Cultural evolution can be seen as a darwinian process where variation, transmission, and selection lead to adaptation of the culture [146]. Examples of cultural evolution go from skills such as fire control, or tool use, to more abstract ones such as mathematics. Those pieces of knowledge are

taught from generation to generation and potentially improved by individuals, for example seen with the increase in complexity of tools [147]. Culture allows an agent to adapt fast by benefiting from the cumulative adaptation of other agents that are socially transmitted to him. In particular, the knowledge that is transmitted to the agent would be hard to discover by an isolated agent.

In psychology, cultural evolution is primarily studied under the experimental paradigm of transmission chains, where they study the iterative change of information along a chain of participants transmitting it [148, 149].

In AI, cultural transmission can be observed with the use of social partner(s) that allow for faster learning by social interaction: either through imitation [150–152], or more complicated teacher-student interactions [153], or finally through the sharing of information across a population of agents potentially allowing them to explore more broadly, escaping local optima [154, 155].

More recently, with the advent of Large language models, capable of complex social interaction with a lot of human biases, we observe works studying the cultural evolution in groups of LLM agents (or mixed LLM-humans groups) [156] and notably the dynamics of iterated cultural transmission [157, 158].

In artificial life, cultural evolution remains deeply understudied with a few exceptions [159].

0.2.5 Interactions between multiples scales

As stated previously, the different adaptation mechanisms mentioned above operate at different scales. Nevertheless, there are a lot of interactions between those adaptive loops.

First, because some of them emerged from other ones. For example, developmental learning emerged from natural evolution, as the ability to adapt during an agent's lifetime was favored when environments were uncertain and changing at the scale of their lifetime [160, 161]. Indeed, developmental learning allows to adapt to changes that happen at a scale much smaller than the scale at which evolution operates.

Secondly, the different adaptation mechanisms also interact in mutual ways. For example, the improvement of knowledge in cultural evolution can often come from the learning of new knowledge through individual developmental learning, making cultural evolution deeply intertwined with developmental learning. Similarly, we observe feedback loop effects between cultural evolution and evolution, for example with gene-culture coevolution exemplified by the coevolution of milk consumption and the gene responsible for milk digestion [162].

In **Evolutionary reinforcement learning**, works study the usefulness of combining reinforcement learning and evolutionary algorithms [163, 164]. In particular, the evolutionary part can be used as a way to maintain a diversity of behaviors within a population of agents in order to enhance exploration and escape local optima (while the RL

part is often used as a "finetuning"). This is exemplified in works using population-based training with RL [44, 165, 166] including quality diversity works which explicitly search for diversity with evolutionary algorithms and use RL to improve the solutions [167, 168].

In **Meta learning** [169, 171–173] (Fig.9), or learning to learn, studies explore how an outer adaptive loop (that some authors view as analogous to an evolutionary scale) can optimize an inner adaptive loop (that some authors view as analogous to a developmental scale) on a wide distribution of learning tasks. This very general principle often approaches the problem by using the outer loop to meta-learn the parameters of the inner adaptive loop to make it more effective (Fig.8). The parameters that are meta-learned range from simple hyperparameters such as the learning rate of the inner loop, to more complex ones such as the parameters of a class of objective functions [174, 175], or parameters shaping the whole update rule [176]. In particular, the outer loop can meta-learn the parameters of a neural network whose internal dynamic will be able to learn "in context" through the update of its internal state after a stream of examples [177]; exemplified in Large Language Models' ability to adapt "in context" [178]. While (hyper)parameters meta-learning is the most used approach, other works directly explore how algorithms in the form of a graph of modules can be meta-optimized by an outer loop [171].

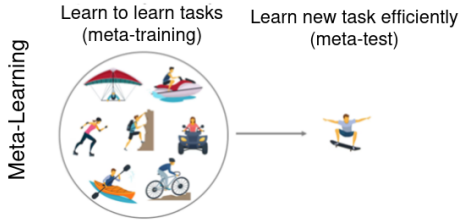


Figure 8: Meta-learning. Fig from [169].

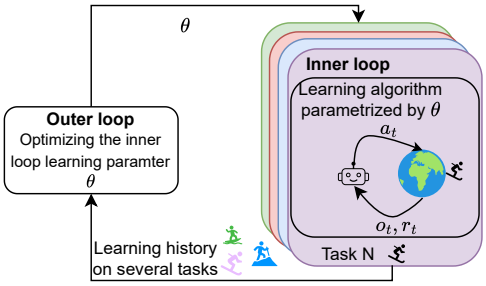


Figure 9: Meta-RL. An outer loop optimizes the parameters of a learning algorithm (inner loop). Fig inspired from [170].

A special case of meta-learning, meta-reinforcement learning [170], explores how efficient learning through trial and error can be meta learned (Fig.8). Examples of this are works studying the meta-learning of (hyper)parameters of a well-defined inner RL algorithm loop. The meta-learned parameters range from effective hyperparameters [179], to reward functions [180], to initial weights of a neural network [181]. Another line of work studies how fast general adaptation can be meta-learned through the learning of parameters of a neural network whose internal dynamics will act similarly to reinforcement learning (in context) [54, 182–185]. More details on meta-RL can be found in Sec.3.1.2.

In particular, works in meta-reinforcement learning also explore the necessary conditions for learning (or learning-related capabilities) to emerge, recognizing the importance of environmental diversity and complexity, for example studying the impact of the variability of the environment on learning [186].

In some aspects, Meta-learning can be considered as a "complexity driven" approach as instead of engineering the adaptive agents (or learning algorithms), meta-learning explores how they can instead be meta-learned and emerge through optimization. This can potentially lead to better learning architecture than those designed by humans [174, 175]. It might also allow never-ending increase in complexity of the learning architecture as the outer loop can still iteratively improve it (compared to fixed algorithms). Meta learning is even considered as a potential pillar toward general intelligence [3].

As we have seen, environmental complexity and dynamics play a central role in shaping agents' adaptation. This is recognized in both Life Science [66] and in Computer Science [187]. We will now explore the importance of the reciprocal interaction between environmental

complexity and agents' adaptation in the bootstrap of an open-ended process.

0.3 Reciprocal causation between environmental complexity and adaptive mechanisms

Having introduced different levels of adaptation and their interactions in the previous section, we will now explore the reciprocal interplay between agent adaptation and environment dynamics. Each section will again draw parallels between existing frameworks in life science (biology, ecology, and cognitive science) and computer science (artificial life and artificial intelligence).

0.3.1 Environmental complexity as a main driver of adaptations

As seen above, how agents adapt at multiple scales is shaped by the structure of their interaction with the environment. This is especially clear in natural evolution where selection is seen as a direct consequence of the environmental pressures and opportunities on the populations. For example, warmer environments induce a pressure on plants that might favor heat resistant capabilities such as storing water [188, 189], or an environment with tall trees might favor taller animals in order to attain the leaves. Comparably, in development and culture, the direct environment is also known to be critical to the "developmental" path an individual takes whether it is through gene activation that depends on the environment, developmental learning of specialized skills or knowledge, or through being taught by social peers.

In particular, the field of human behavioral ecology (HBE) explores the ecological drivers for certain traits of humans including the emergence of agriculture or the increase in brain capacity [190–192]. For example, some works hypothesize the important role of environmental variability and instability on human evolution [190, 193], potentially responsible for the evolution of human general capabilities.

In AI and especially the field of reinforcement learning, the environment is known to be critical to the agents' final capabilities as the agent will learn based on its interaction with it. In particular, environments with more complex dynamics might require a higher level of cognition to be solved, like some form of reasoning or even specialized skills such as learning to read [194].

As well as the environment complexity, the overall distribution of tasks an agent is trained on has been shown to be critical. It has been shown that training on a wide diversity of environments (ideally open-ended) can lead to general skills [44, 45, 195, 196]. In particular, to cope with variable environments, the agent's training might meta-learn a fast learning adaptation mechanism, allowing the agent to

explore and adapt during its lifetime to the environment that might be unknown or new [182–184, 186]. In fact, works have shown that depending on the variability and complexity of the task, the adaptation might learn either a fast learning mechanism or innate behavior [186]. In constant environments, innate behavior often suffices and proves more efficient. However, in environments with moderate variability, general fast-adapting agents are favored, as they can explore and quickly learn the dynamics of the current environment—albeit at the cost of dedicating time to this adaptive process.

In addition to the distribution of environments being important, the sequence of environments is also crucial for efficient adaptation processes. In particular, RL training often uses curriculum learning techniques which start from easier environments and progressively expose agents to more and more complex environments. This has been shown to lead to better performance than training randomly on the distribution of environments [195, 197–199]. In practice, in curriculum learning, the next environments to train on might be selected according to the current capabilities of the agent, learning progress, or uncertainty about the task. We refer to [198] for an overview of curriculum learning.

In Artificial Life (Alife), much of the research focuses on discovering environmental dynamics that foster the emergence of complexity, often through simulated ecosystems incorporating evolutionary processes [5, 32–38, 42, 91] (Fig.10). Traditionally, these studies have explored relatively simple foraging environments, where agents develop control strategies to gather regenerating resources. Despite their simplicity, such environments can still give rise to complex phenomena, such as altruistic behaviors [200], most often due to their multi-agency nature that we will cover in Sec.0.3.3.

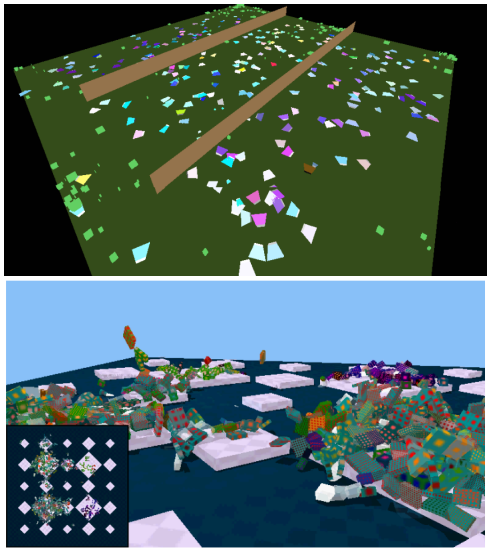


Figure 10: **Artificial life ecosystems** with evolutionary processes. (Top) Polyworld figure from [34]. (Bottom) Division block figure from [38]..

More recently, some works in Alife investigated environments of greater complexity, which provide richer opportunities for adaptation and the emergence of intricate behaviors. These efforts have enabled the exploration of more complex dynamics and adaptive strategies [50, 201].

The role of environments in RL or Alife can be linked to the idea of a complexity-driven approach. Instead of hand-engineering certain cognitive processes, the idea is to train an agent on an interesting (and potentially diverse set of) environments and expect interesting capabilities to be learned or emerge. In addition, the environment variability, used in ML works aiming for general skills or meta-learning, echoes with the variable environment studied in human behavioral ecology [190, 193].

0.3.2 Reciprocal causation between environmental structure and agent's adaptability

Not only does the agent adapt to the environment, but the environment itself also changes according to the agents' behavior. This influence from the agent to the environment is called **nich construction**[203, 205–207] or **ecosystem-engineering** [208] and covers sim-

- "**Organisms not only adapt to environments, but in part also construct them** [202]. Hence, many of the sources of natural selection to which organisms are exposed exist partly as a consequence of the niche constructing activities of past and present generations of organisms."[203]
- "'**Reciprocal causation' captures the idea that developing organisms are not solely products, but are also causes, of evolution.**" [204]

ple cases such as plants producing oxygen as a byproduct of their activity, to more complicated niche construction such as agriculture. These modifications of the environment due to the agents' actions might have an impact on the environmental pressures and opportunities, and in particular on the agent's own fitness. Notably, an agent might evolve or learn to change the environment for potential benefits, for example stabilizing and increasing its food supply through agriculture or building nests for protection. However, niche construction can also be detrimental, for example with agents dumping their detritus in their environment or overconsuming their own resources leading to their depletion.

Such agent-induced changes in the environment might impact the agent itself or even its offspring as the environment will also be passed to the next generation (in addition to the genes) through **ecological inheritance** [203] (Fig.11). In addition, ecosystem engineering might have an impact on other agents that share the environment. For example, cyanobacteria produced oxygen that changed the ecosystem' opportunities and pressures and may have led to the evolution of complex, bigger multicellular life forms [210].

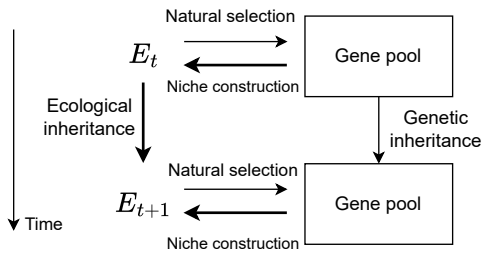


Figure 11: Feedback loop effects between agent and environment adaptation. Fig from [209].

Niche construction has been for a long time disregarded as a minor mechanism driving evolution but recently regained attention viewing it as an important evolutionary mechanism, even termed "the neglected process in evolution" [203]. In particular, thinking about evolution in a broader sense taking into account niche construction is developed in the *Extended evolutionary synthesis* (EES) [204]. EES extends classical evolutionary theory which often only takes into account a one-way causal effect from environment to agent evolution. Notably, EES emphasizes the importance of the developmental trajectory of agents in natural evolution. In particular, it emphasizes the fact that agent development – from gene expression depending on environmental context, to complex developmental learning and ecosystem engineering – will affect its selection as well as its environment (Fig.12).

- **Niche construction**: "informed activities of organisms that influence the environment and affect the fitness of the population." [211]
- **Ecological inheritance**: "the persistence of environmental modifications by a species over multiple generations to influence the evolution of that or other species." [211]

As niche construction leads to new pressures and opportunities from the environment, it can therefore potentially lead to new adaptations from the agents, which in turn might also again modify the environment, resulting in novel selective pressures. This reciprocal causation between agent and environment can lead to an interesting feedback loop effect, where agent adaptation is shaped by the environment but the potentially resulting new behavior might as well change the environment again. These **eco-evolutionary** dynamics can potentially lead to never-ending complexity by constantly setting new problems and opportunities, and in this way be a central driver of open-endedness in the living world.

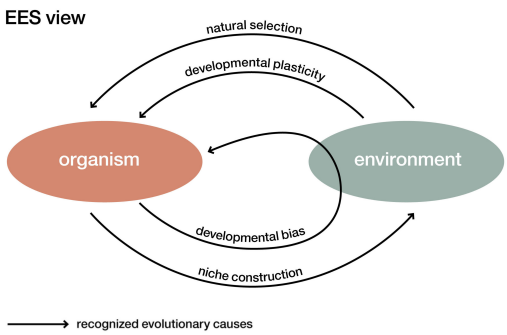


Figure 12: Extended evolutionary synthesis (EES) scheme : *https://extendedevolutionarysynthesis.com/about-the-ees/*

In AI however, methods such as RL or evolutionary algorithms often typically rely on an episodic training paradigm, where the environment is regularly reset to an initial configuration (when the task is solved or after a predefined time). This allows training on a stable specific version of the environment but, in turn, breaks the potentially reciprocal causation between the environment and agents (as potential changes are reset). Few exceptions exist advocating for non-episodic training, where the environment isn't reset [212].

Curriculum learning in machine learning, where the next environment is chosen (or procedurally generated) according to the agent's current capabilities, can also lead to "co-evolution" of agents and environments [195, 197–199, 213]– as the agent's adaptation will drive the "environmental trajectory" and, reciprocally, the "environment trajectory" will drive the agent's adaptation. To some extent, this mechanism can be partly compared to the eco-evolutionary dynamics.

In Alife, works on artificial ecosystems often do not reset the environment and let it evolve for an extended period of time [6, 71, 72, 85–87]. However, environments used in Alife are often very simple, which might not allow the feedback loop effects to have strong visible dynamics. In fact, feedback loop effects between agents' adaptation and their environment can create unstable dynamics, potentially resulting in population collapse—for instance, when agents deplete all available resources. This may explain the tendency to simplify or omit such feedback mechanisms.

0.3.3 Multi agency as a driver of open-endedness

In the case of multi-agent systems, one cause of environment variability is the adaptation of other agents (as from a single-agent perspective, other agents also constitute the environment). Multi-agency is in fact a potential cause of "continual problem generation," where each agent might have to continuously re-adapt to the changing behavior of others, changing again the behavior of agents, ultimately leading to feedback loop effects. These co-adaptations can lead to a continual increase in complexity of the system for example through arm race like dynamic where agents continuously improve to beat (or adapt to) the others. In fact, even environments that are very simple in their single-agent version can lead to very complex behavior when multiple agents are involved. To illustrate this, Leibo et al [214] take the example of the game of Go, which consists of "capturing" the biggest territory on a grid by surrounding areas with rocks. With a single player, this would trivially consist of placing the stone on the edge of the grid. However, when played by two adversarial players this leads to very complex strategies that have been continuously improved over thousands of years.

These dynamics of agents co-adaptation have been observed in natural evolution as well, for example with coevolution of different species where adaptation of one can lead to new pressure for the other [215, 216]. This need for continual adaptation as a response to other species' adaptation has been theorized in the red queen hypothesis [217]. Examples of such dynamics can be seen in predator-prey systems such as gazelle and cheetah which coevolve to be faster than the other, exemplifying arms race [215], or even plants evolving defenses that are countered over and over by herbivores [218–220], or parasite-host relationships [221]. The co-adaptation of individuals can also be seen in the same species through development and culture, for example with literal arms race between groups of individuals in humans.

In simulation, several studies have shown arm race like dynamics as a driver of continual increase in complexity. For example, some RL

works show a continual increase in strategies' complexity emerging from adversarial play where opposing teams of adaptive agents constantly learn strategies and counter-strategies to beat the other team [115], notably with the example of emergent tool use in a simple hide and seek scenario [39]. Other RL works also use arm race like dynamics through self-play [222] – training agents against themselves (or a previous version of themselves) in a multi-agent scenario – allowing to reach superhuman performances in games such as chess, go, or video games, etc. [109, 113, 114, 223, 224].

In a more asymmetrical relationship between agents, several works in AI consider the co-adaptation between a problem setting agent and a problem solver agent to achieve an increase in complexity [48, 49, 65, 195]. This also includes works on generative adversarial networks (GAN) [225] which also use this dynamic of arms race by using two agents in competition, having a generator agent generating "fake" data, which the discriminator agent has to discriminate against true data, with the ultimate goal of training a generator agent able to generate data that is hard to discriminate from true data.

Alife works often consider multi-agent interactions as an important driver of emergent complexity, where shifts in equilibrium are due to other agents' adaptations. This includes works displaying multi-species simulations effectively reproducing some aspects of natural evolution [5, 32–38, 42, 91] including arm race dynamics [226]. This also includes works where macro-individuals (or dynamics) emerge from the self-organization of several simple atomic constituents which strongly influence each other [7, 75, 227].

In addition to competition, multi-agency can also lead to an emergent increase in complexity through cooperation of groups of individuals with division of labor, better exploration as well as information sharing and culture. Examples in natural evolution are mutualistic coevolution, where each species benefits from the coevolution, such as flowers-pollinators coevolution with plants evolving to attract pollinators and pollinators evolving to find the plant, eat the nectar and transport the pollen [228, 229] .

0.3.4 Conclusion

In this section, we've explored the **importance of environment in shaping the adaptation** of an agent. We've further explored this interaction showing that it is in fact a two-way interaction where agents adapt to the environment but also where **environment is changed due to agent activities**. This **interplay between agent and environment adaptation, leading to feedback loop effect**, is in fact a great candidate for implementing systems with continual increases in complexity. In addition, we presented how multi-agent dynamics, through the interactions between different co-adaptive agents, can also lead to feedback loop effects with increases in complexity, for example, through competition and arms races.

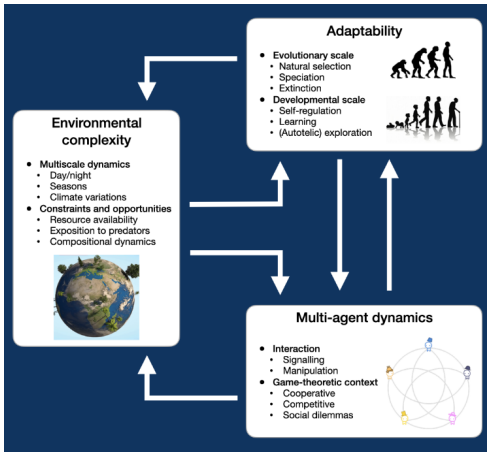


Figure 13: Interplay between agent adaptation, environmental complexity and multi agent dynamics through feedback loop effects in the Origins framework [230]

0.4 Objectives and Contributions

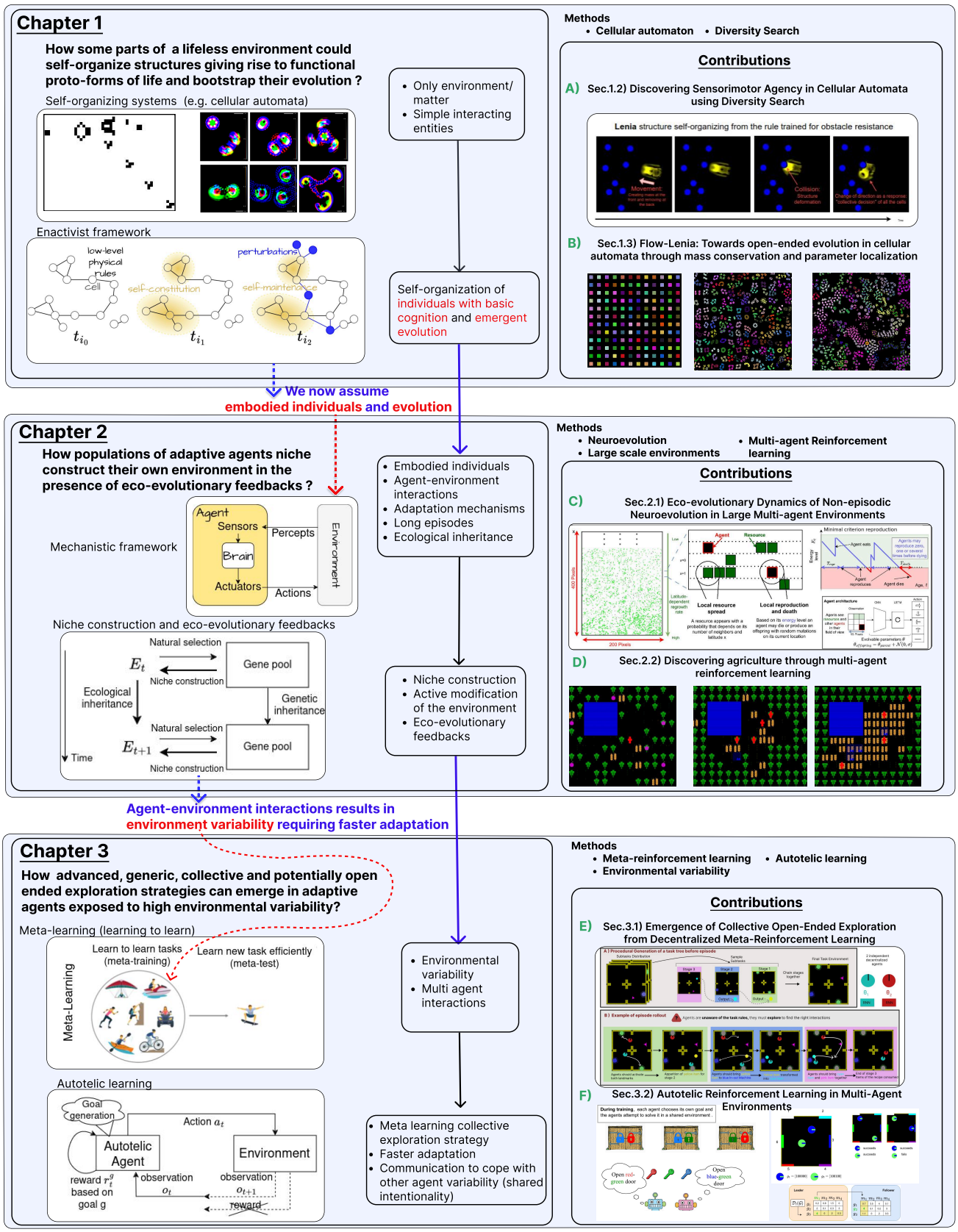


Figure 14: Overview of the contributions of this thesis. The thesis follows a progressive structure where the emergent behaviors derived in earlier chapters form the basis (or initial state) for subsequent investigations.

So far, we have seen that the interplay between agent adaptation and environmental dynamics offers a promising foundation for constructing systems with open-ended dynamics. In this thesis, we investigate, through in silico experimentation at various levels of abstraction, how interactions between complex environments and adaptive entities can drive emergent complexity.

We adopt a bottom-up approach, wherein each contribution isolates certain mechanisms that we believe are key drivers of emergent complexity within a system. By focusing on **isolated** specific assumptions and initial states conducive to emergent complexity, we aim to elucidate these mechanisms, ensuring the analysis is stable, controlled, and comprehensible.

The thesis follows a progressive structure where the emergent behaviors derived in earlier chapters form the basis (or initial state) for subsequent investigations. In the final discussion (Chap.5), we synthesize these findings and the links between them and try to outline a potential framework for open-ended simulations capable of fostering general agents.

This progression is depicted in Fig.14 and summarized below:

In chapter 1 , we consider a lifeless environment, in an initial state where there is literally *no body* (and thus no sensing, no acting, no agent, no evolution). Our objective is to study **how some parts of such a lifeless environment could self-organize structures giving rise to functional proto-forms of life and bootstrap their evolution**. For this aim, we rely on state-of-the-art continuous cellular automata, using diversity search algorithms to explore their parameter space in the search of relevant self-organizing phenomena.

In a first contribution (Fig.14.A), we apply diversity search and curriculum learning algorithms in a continuous cellular automata for the search of system rules leading to the systematic emergence of self-organizing structures displaying basic sensorimotor capabilities – i.e. proto-agents that can react to perturbations of the environment. Interestingly, we discover self-organized structures able to seemingly take decisions at the macro scale only from the collective dynamics of many atomic parts, i.e. without any notion of a central "brain", sensors and actuators. In addition, these self-organized agents show impressive generalization capabilities to conditions not seen during the search.

Our second contribution (Fig.14.B) delves into the self-organization of evolutionary dynamics within a similar lifeless environment. In particular, we extend the continuous cellular automata used in the previous contribution, allowing us to introduce "multi-species" simulations – where self-organizing structures governed by different update rules can coexist. In these multi-species simulations, we observe evolutionary dynamics occurring due to the physics of the system without any external evolutionary algorithm. In particular, we observe an emerging evolutionary activity resulting from the cooperative or competitive interactions between diverse self-organizing "species", here again only from the collective dynamics of many atomic parts.

Chapter 1 thus demonstrates the emergence of individuality, basic cognition (in the form of sensorimotor capabilities) and evolution in an originally lifeless environment through interactions among simple atomic elements. This ultimately results in environments with **adaptive agents, well separated from the environment,** which proliferate and die through interactions with neighboring entities. In the rest of the thesis, we then consider a more standard dichotomy between an environment and agents interacting with it, pre-equipped with sensors, actuators, and decision-making abilities.

In chapter 2 , we study the interplay between agent adaptation and environment dynamics with embodied agents well separated from the environment. More precisely, we explore emergent eco-evolutionary dynamics and niche construction phenomena, asking: **How populations of adaptive agents niche construct their own environment in the presence of eco-evolutionary feedbacks ?** . We focus on two other contributions.

The first one (Fig.14.C) presents a system where agents continuously evolve without any environment or population reset, enabling eco-evolutionary feedback. The environment is a large grid world with complex spatiotemporal resource generation, containing many agents that are each controlled by an evolvable recurrent neural network and locally reproduce based on their internal physiology. We show that neuroevolution can operate in an ecologically valid non-episodic multi-agent setting, finding sustainable collective foraging strategies in the presence of a complex interplay between ecological and evolutionary dynamics.

The second contribution (Fig.14.D) investigates the emergence of agricultural practices within a population of reinforcement learning agents. Situated in an environment with different resources that are in competition with each other, these agents learn to eco-engineer their surroundings to promote the proliferation of beneficial resources. This convergence toward collective niche construction strategies underscores the agents' ability to modify their environment to their advantage.

Chapter 2 therefore introduces niche construction and eco-evolutionary feedback. In particular, this complex interplay between adapting agents and the environment can lead to environments with fast variations that the agents have to deal with. We will further explore in chapter 3 the impact of this variation on agents' adaptation.

In chapter 3 , we control the environmental variability and study **How advanced, generic, collective and potentially open ended exploration strategies can emerge in adaptive agents exposed to high environmental variability?** .

Our first contribution (Fig.14.E) leverages procedurally generated hierarchical tasks to study the emergence of collective exploration in a group of independent agents. From the training on a diverse distribution of tasks where the underlying rules have to be discovered,

agents meta-learn to collectively explore the affordances of the environment. The agents also show interesting generalization to new tasks and longer chains of tasks (with more objects, etc.) not seen during training.

In the second contribution (Fig.14.F), we shift to groups of independent agents with a predefined autotelic exploration mechanism and study the emergence of shared intentionality to cope with variability induced by other exploring agents. In particular, from independent reward maximization, the agents learn to communicate and align their goals, ultimately achieving more effective learning compared to agents independently sampling their goals.

This third and last contribution chapter demonstrates how collective exploration and shared intentionality can emerge as effective mechanisms for coping with highly variable environments, particularly when agents need to coordinate their learning and discovery processes.

0.4.1 List of contributions

The work presented in this thesis is based on the following publications as well as accompanying codebases [2] and other materials. Stars next to author names indicate co-first authors.

2: all of our publications are accompanied with open-source code to reproduce the results and analysis

Publications

- Hamon*, G., Etcheverry*, M., Chan, B. W. C., Moulin-Frier, C., Oudeyer, P. Y. (2024). *Discovering Sensorimotor Agency in Cellular Automata using Diversity Search.* arXiv preprint arXiv:2402.10236.
  Under review at Science Advances.
  Contribution A, sec.1.2.
  Preprint, Code, Blogpost.

- Plantec, E., Hamon, G., Etcheverry, M., Oudeyer, P. Y., Moulin-Frier, C., Chan, B. W. C. (2023). *Flow-Lenia: Towards open-ended evolution in cellular automata through mass conservation and parameter localization.* **In Artificial Life Conference Proceedings** 35 (Vol. 2023, No. 1, p. 131). MIT Press.
  Best paper award at Alife 2023 and **Extended version accepted in the Alife Journal.**
  Contribution B, sec.1.3.
  Paper,Code,Base Companion website.,Second Companion website.
  Video for the Virtual Creatures Competition 2024 (VCC2024).

- Hamon*, G., Nisioti*, E., Moulin-Frier, C. (2023, July). *Eco-evolutionary dynamics of non-episodic neuroevolution in large multi-agent environments.* **In Proceedings of the Companion Conference on Genetic and Evolutionary Computation** (pp. 143-146).
  Contribution C, sec.2.1.
  Paper, Preprint, Code,Companion website.

- Bornemann*, R., Hamon*, G., Nisioti, E., Moulin-Frier, C. (2023) *Emergence of collective open-ended exploration from Decentralized Meta-Reinforcement learning.* **In Second Agent Learning in Open-Endedness (ALOE) Workshop at Neurips 2023**.
  Contribution E, sec.3.1.
  Preprint, Code,Companion website.

- Nisioti*, E., Masquil*, E., Hamon*, G., Moulin-Frier, C. (2023). *Autotelic Reinforcement Learning in Multi-Agent Environments.* **In Conference on Lifelong Learning Agents** (pp. 137-161). PMLR.
  Contribution F, Sec.3.2.
  Preprint, Code,

- Léger*, C., Hamon*, G., Nisioti, E., Hinaut, X., Moulin-Frier, C. (2024). *Evolving Reservoirs for Meta Reinforcement Learning.* **In International Conference on the Applications of Evolutionary Computation** (Part of EvoStar) (pp. 36-60). Cham: Springer Nature Switzerland.
  Sec.4.2
  Paper, Preprint, Code

- Taylor-Davies, M., Hamon, G., Boulet, T., Moulin-Frier, C. (2024). *Emergent kin selection of altruistic feeding via non-episodic neuroevolution.* arXiv preprint arXiv:2411.10536.
  Accepted at The International Conference on the Applications of Evolutionary Computation (EvoAPPS) 2025 (part of EvoStar).
  Sec.4.1
  Preprint, Code

Blogpost

- Hamon*, G., Etcheverry, M., Chan, B. W. C., Moulin-Frier, C., Oudeyer, P. Y. (2022). *Learning sensorimotor agency in cellular automata.*
  `https://developmentalsystems.org/sensorimotor-lenia/`

Works in progress Still works in progress but papers will be written.

- *Discovering agriculture through multi-agent reinforcement learning.*
  Began with a 3 month visit in Ricard Solé's Complex system Lab, Universitat Pompeu Fabra, Barcelona, Spain.
  Contribution D, sec.2.2.

- *Meta-learning curiosity through reward maximization in a variable compositional environment.*
  More information and Preliminary results in Sec.4.4.

Code In addition to the code for all the papers, we also released the code for a fast transformer and RL library in JAX. The implementation follows the paper "Stabilizing Transformers for Reinforcement Learning" from Parisotto et al [231]. We achieve state-of-the-art performances on the challenging craftax RL environment [232]. More information can be found in Sec.4.3.

Popular science

- C Moulin-Frier, G Hamon, PY Oudeyer. (2024). *Quand l'IA explore les prémices de l'évolution.* **Pour la Science.**
  Popular science article in French encompassing our works on cellular automata described in Chap.1. Targeted at a general audience. (Pour la Science is the French equivalent of Scientific American.)

Main contributions

Low level: emergence of basic cognition and open ended evolution

1

How did the first individuals emerge from a lifeless environment of pure matter? And how did these primordial entities bootstrap the process of open-ended evolution? While evolutionary theory provides powerful frameworks for studying ongoing evolution, the question of its very origins—the transition from non-life to life—remains one of science's most fascinating challenges.

In this chapter, we explore how basic cognitive agents and evolutionary processes might emerge from the simplest of starting conditions: an environment with no notion of individual, containing only matter and fundamental physical rules. Our investigation focuses on self-organization—the spontaneous emergence of macroscopic structures whose properties transcend those of their microscopic components. Just as water molecules can self-organize into intricate snowflakes or stars into spiral galaxies, we seek to understand how matter might self-organize into the first proto-living individuals with self-maintenance and basic "cognition". Ultimately, we aim to unravel how these self-organizing systems could initiate an evolutionary process with mutation and selection.

Our approach is grounded in the enactivist framework [81, 82] (Fig.1.1), which considers that every part of the body of an organism participates in cognitive processes and that the individual must come to existence from the self-organization of its parts. This stands in contrast to the mechanistic framework (Fig.1.2), which already assumes the pre-existence of a body and a central "brain" well separated from the environment, interacting with it through predefined sensors and actuators. The mechanistic view, which is largely dominant in AI, already presupposes an agent embodiment and rather focuses on understanding how higher-level cognitive interactions can arise. In the enactive view, "the question of the bodily constitution is conceptually prior to any particular functional account of a cognitive subsystem".

To investigate these questions empirically, we employ cellular automata as our environment testbed following a long list of works using it to study self-organization of basic cognition, minimal criterion for life, artificial life, autopoiesis [77–80]. Cellular automata are in their classic form, grid of "cells", which are sequentially updated through a local rule, local in the sense that only the neighbouring cells are taken into account to compute the update. Cellular automata are complex system where simple local rules can often lead to very complex self-organizing patterns and interactions [233] like the game of Life. See Sec.1.1.1 for more details on celluar automata.

In our investigation, we utilize Lenia, a parametrized class of continuous cellular automata where each parameter set defines a distinct set of rules —including, for specific parameters, the well-known Game of Life. Lenia has demonstrated remarkable capacity for generating diverse, life-like patterns [8, 9], making it an ideal framework for study-

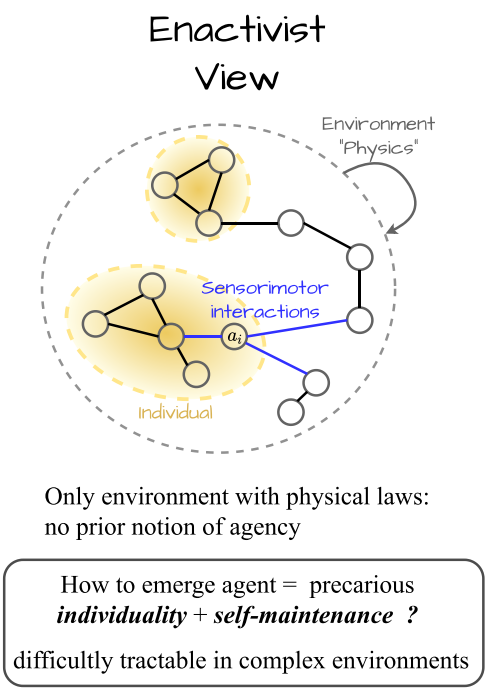
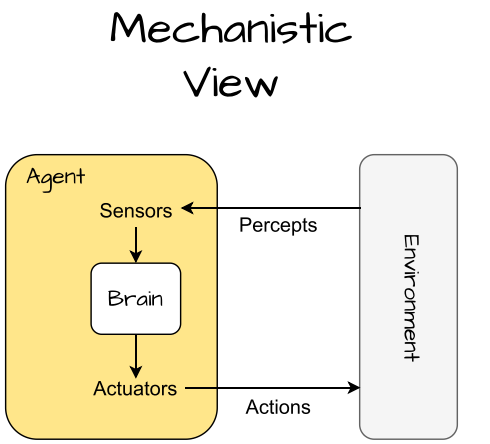


Figure 1.1: Enactivist framework

ing the self-organization of individuals with basic cognitive capabilities. More details on Lenia is given in section.1.1.2.

In the first section, we explore the emergence of basic cognition in the form of sensorimotor capabilities—specifically, how organized responses to environmental perturbations can arise. Using advanced AI techniques such as diversity search [167, 168, 234, 235], we explore the vast space of Lenia parameters in the search of self-organizing patterns resembling proto-individuals capable of coherent sensorimotor capabilities (Fig.1.3). Remarkably, the behavior is observed at the level of the macro individual in a coherent manner without the existence of a central brain but only from the coherent self-organization of thousands of simple parts. Moreover, these systems display impressive generalization capabilities, responding robustly to novel environmental conditions not encountered during the parameter search.

In the second section, we explore the question of the self-organization of open-ended evolution. For this we introduce a mass-conservative extension of Lenia: Flow Lenia. Mass conservation, a fundamental constraint of biological systems [236], is interesting in such simulation as it makes it easier to introduce environmental pressures in the system (such as a need for resources), as well as lead to more stable spatially localized patterns. Most importantly, this change of the system also allows for the introduction of heterogeneous local rules, where parameter of the update rule are embedded in the system dynamic, allowing for different "species" to coexist in the same simulation. "Multi-species" simulations from the competition for matter lead to **selective pressure akin to evolution, just from the physic of the system** (Fig.1.4). The emergence of "evolution" in this "only environment" scenario also shows the complex environment variation that can arise from interaction between self-organized individuals and the rest of the environment.



One assumes the pre-existence of a body with sensors and actuators, through which an agent can interact with its environment

Figure 1.2: Mechanistic framework

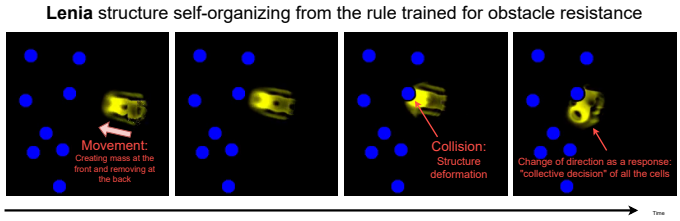


Figure 1.3: Robust moving structure emerging from the rules of the environment that are discovered by our method in the first contribution Sec.1.2. Blogpost with demo and videos *https://developmentalsystems.org/sensorimotor-lenia/*

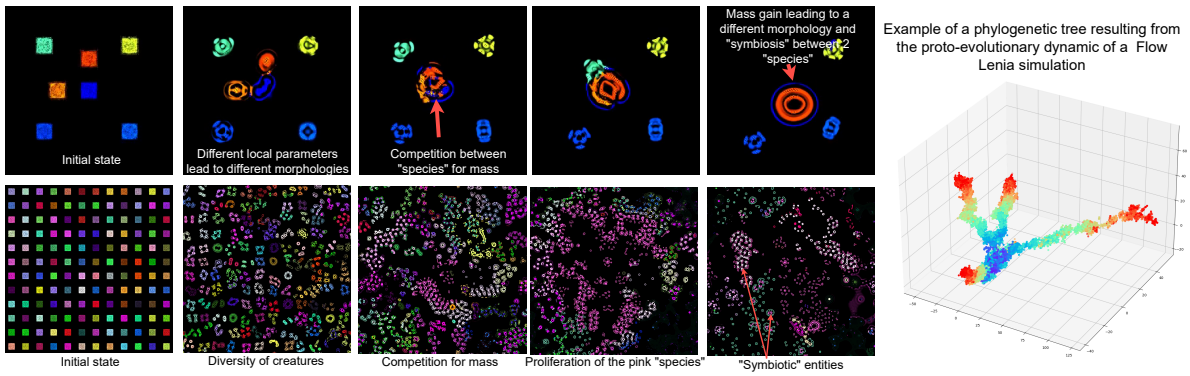


Figure 1.4: Interactions between patterns in a multi species simulation in Flow Lenia. Video at *https://sites.google.com/view/flowlenia/* and at this link

Contributions associated with this chapter

- Hamon*, G., Etcheverry*, M., Chan, B. W. C., Moulin-Frier, C., Oudeyer, P. Y. (2024). Discovering Sensorimotor Agency in Cellular Automata using Diversity Search. arXiv preprint arXiv:2402.10236. Under review at Science Advances.
  Sec.1.2.
- Plantec, E., Hamon, G., Etcheverry, M., Oudeyer, P. Y., Moulin-Frier, C., Chan, B. W. C. (2023). Flow-Lenia: Towards open-ended evolution in cellular automata through mass conservation and parameter localization. In Artificial Life Conference Proceedings 35 (Vol. 2023, No. 1, p. 131). MIT Press.
  Best paper award at Alife 2023 and extended version accepted in the Alife Journal.
  Sec.1.3.
- Hamon*, G., Etcheverry, M., Chan, B. W. C., Moulin-Frier, C., Oudeyer, P. Y. (2022). Learning sensorimotor agency in cellular automata. Blogpost associated with Discovering Sensorimotor Agency in Cellular Automata using Diversity Search.
- C Moulin-Frier, G Hamon, PY Oudeyer. (2024). Quand l'IA explore les prémices de l'évolution. Pour la Science.
  Popular science article in french encompassing our works on cellular automata described in this chapter. Destined to a general audience. (Pour la Science is the french equivalent of Scientific American.)

1.1 Cellular automata and Lenia

In this section, we present the cellular automata framework (Sec.1.1.1) as well as the Lenia cellular automaton [8, 9] (Sec.1.1.2) that will be used in the rest of this chapter.

1.1.1 Cellular automata

Cellular automata (CA) are, in their classic form, a grid of "cells" $A = \{a_x\}$ that evolve through time $A^{t=1} \longrightarrow ... \longrightarrow A^{t=T}$ via the same local "physics-like" laws. More precisely, the cells sequentially update their state based on the states of their neighbours: $a_x^{t+1} = f(\mathcal{N}(a_x^t))$, where $x \in \mathcal{X}$ is the position of the cell on the grid, a_x is the state of the cell, and $\mathcal{N}(a_x^t)$ is the neighbourhood of the cell (including itself). The dynamic of the CA is thus entirely defined by the initialization $A^{t=1}$ (initial state of the cells in the grid) and the update rule f (how a cell updates based on its neighbours). But predicting the system long term behavior is a difficult challenge, even for simple rules, due to their potential chaotic dynamics [233].

Cellular automata (CA), notably Conway's Game of Life (GoL) [68] (Fig.1.5), have attracted a lot of interest from the artificial life (ALife) community because of the emergence of life-reminiscent spatially-localized patterns (SLPs). These patterns are of special interest as instances of autopoietic structures (i.e self-produced and self-maintained structures) [77], a fundamental property of life and cognition as proposed in Maturana and Varela theory [67].

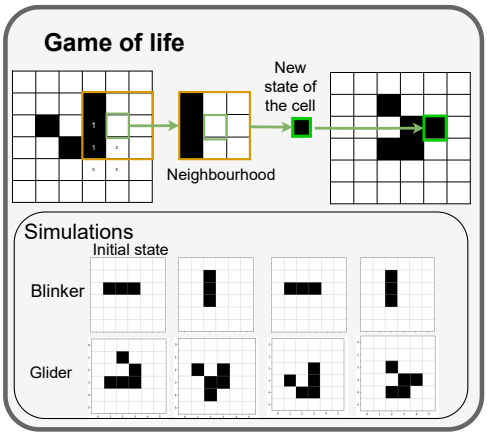


Figure 1.5: Conway's Game of Life cellular automata[68]. Each cell is updated based on its neighbourood. The self-organization of a collective of simple cells can enable the emergence of localized macro structures, such as the "blinker", and sometimes mobile ones like the "glider."

1.1.2 Lenia cellular automaton

Lenia [8, 9] is a class of continuous cellular automata where each CA instance is defined by a set of parameters θ that conditions the CA rule f_θ. Once the parameters θ conditioning the update rule have been chosen, the system is a classical CA where the initial grid pattern $A^{t=1}$ is iteratively updated. Previous works in Lenia have shown that there exist local update rules f, that can lead to the self-organization of long-term stable complex patterns that display interesting diverse life-like behaviors [8, 9, 235], as shown in Figure.1.6.

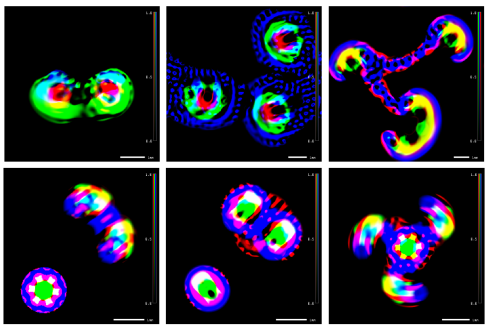


Figure 1.6: Example of Lenia patterns from [9]. Video at `https://www.youtube.com/watch?v=HT49wpyux-k`

In the multi-channel version of Lenia [9], the system is composed of several communicating grids which we call channels. Intuitively, we can see channels as the domain of existence of a certain type of cell. Each type of cell has its own physics : it has its own way to interact with other cells of its type (intra-channel influence) and also its own way to interact with cells of other types (cross-channel influence).

The update of a cell $a_{x,c}$ at position x in channel c can be decomposed in three steps, illustrated in Figure.1.7 (and animated in `https://developmentalsystems.org/sensorimotor-lenia/#lenia`). First, the cell senses its neighbourhood in some channels (its neighbourhood in its channel, with cells of the same type but also in other channels with other types of cells) through convolution kernels which

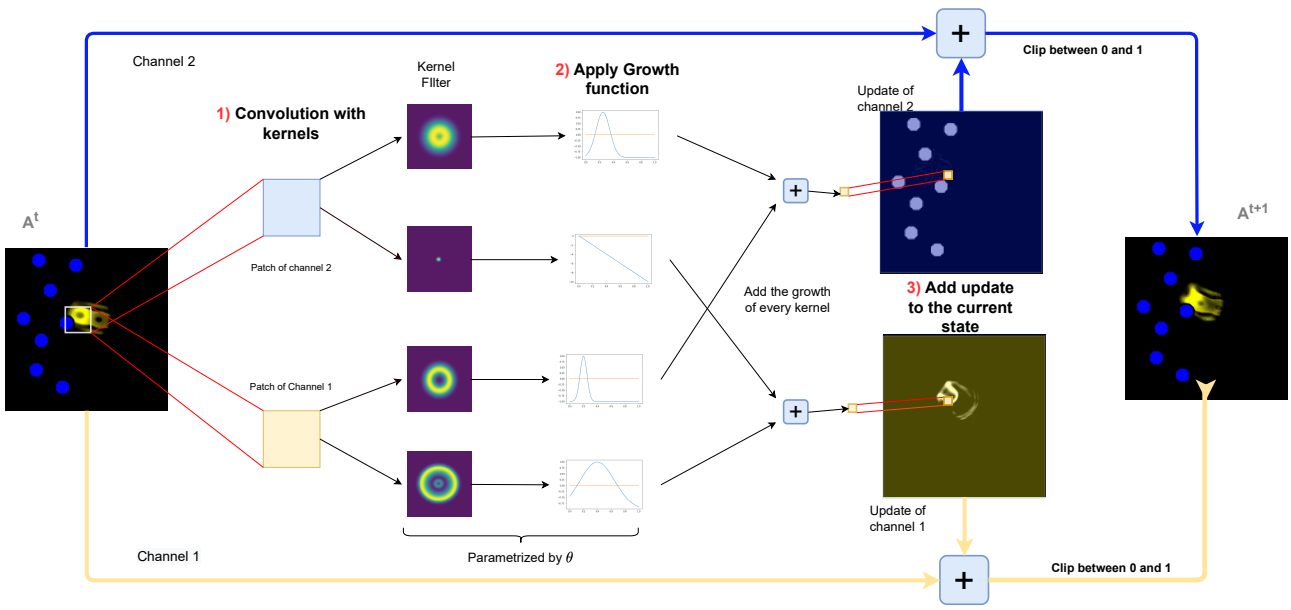


Figure 1.7: Detailed view of a step in Lenia (multi-channel). 1) A convolution with the kernels followed by 2) a growth function is applied on each channel, resulting in a growth update which is 3) added to the current state. Both the convolution kernels and the non- linear growth function in the learnable channel are parameterized

are filters K_k of different shapes and sizes. Second, the cell converts this sensing into an update (whether positive or negative growth or neutral) through growth functions G_k associated with the kernels. Finally, the cell modifies its state by summing the scalars obtained after the growth functions and adding it to its current state. After the update of every rule has been applied, the state is clipped between 0 and 1. Each (kernel,growth function) couple is associated to the source channel c_s it senses, and to the target channel c_t it updates. A couple (kernel, growth function) characterizes a rule on how a type of cell c_t reacts to its neighbourhood of cells of type c_s. Note that c_s and c_t could be the same, which corresponds to interaction of cells of the same type (intra-channel influence). Note also that we can have several rules, i.e. several (kernel,growth function) couples, characterizing the interaction between c_s and c_t. We refer to section 1.3.2 for a more detailed description of the update rule with formulas.

1.2 Discovering Sensorimotor Agency in Cellular Automata using Diversity Search

Context

This contribution results from a collaboration with Bert Chan (google deepmind Tokyo) in 2021-2022. It has been submitted to Science Advances (and is currently under review):

- Hamon*, G., Etcheverry*, M., Chan, B. W. C., Moulin-Frier, C., Oudeyer, P. Y. (2024). *Discovering Sensorimotor Agency in Cellular Automata using Diversity Search.* arXiv preprint arXiv:2402.10236. Paper, Code

I am co-first author of this article.

A blogpost associated with the publication, with videos and interactive demo, is also available at `https://developmentalsystems.org/sensorimotor-lenia/`.

- Hamon*, G., Etcheverry, M., Chan, B. W. C., Moulin-Frier, C., Oudeyer, P. Y. (2022). *Learning sensorimotor agency in cellular automata.* Blogtpost link.

We strongly encourage to take a look at the blogpost to get a better view of the dynamic of the system with videos.

Note that, as this contribution was submitted to an interdisciplinary journal, its structure is modified with the material and methods at the end of it Sec.1.2.4.

Abstract

The field of Artificial Life studies how life-like phenomena such as agency and self-regulation can self-organize in computer simulations. In cellular automata (CA), a key open-question is whether it is possible to find environment rules that self-organize robust "individuals" from an initial state with no prior existence of things like "bodies", "brain", "perception" or "action". Here, we leverage recent advances in machine learning, combining algorithms for diversity search, curriculum learning and gradient descent, to automate the search of such "individuals". We show that this approach enables us to systematically find environmental conditions in CA leading to self-organization of basic forms of agency, i.e. localized structures that move around and react in a coherent and highly robust manner to external obstacles, maintain their integrity, and have strong capabilities to generalize to new environments. We discuss how this approach opens new perspectives in AI and synthetic bioengineering.

1.2.1 Introduction

Understanding how life, cognition and natural agency emerged has been a central debate across many sectors of life sciences. Biological organisms are made of collections of cells that follow low-level distributed rules and yet they constitute a coherent unitary whole, displaying strong *individuality*[1] and *self-maintenance*[2] in their environment, what was described to be an *autopoietic system*[3]. While a central concept in theoretical biology, the characterization of an autopoietic system and the understanding of the processes underlying its self-organization remain a live issue. Further demystifying how those processes do not just give rise to organic individuation[4] but also to sensorimotor[5] and even intersubjective[6] agency, is at the center of the debate [238]. In fact, recent advances in biology and basal cognition suggest that many autopoietic systems that we find in nature, including plants and brainless animals, are robust sensorimotor agents capable of using a body for sensing opportunities, computing decisions and acting in their environment [239]. The pragmatic and complementary question to the debate, central in artificial life (ALife) and artificial intelligence (AI) research, is: can we engineer the necessary ingredients leading to the emergence of functional forms of life and sensorimotor agency in an artificial substrata in which initially there is literally no body (and thus no sensing, no acting, no agent)? Although there is already a large body of work that proposes to study the emergence of life and cognition in agents-as-they-could-be, it is generally done either by jumping over the biological processes that enable organisms to survive (the *mechanistic view*, as in e.g. reinforcement learning, which considers a pre-existing agent with predefined sensors and actuators) or inconclusive so-far in showcasing higher-level forms of sensorimotor agency (the *enactivist view*, as in e.g. artificial chemistry which studies how some form of agency can emerge from low-level chemical reactions). Herein, after giving some background on the mechanistic and enactivist views on cognition and on their respective limitations, we suggest that modern tools from machine learning (ML) can help us bridge the gap between those two views. Whereas those tools have mainly been deployed within the mechanistic framework, we show that they can efficiently assist the discovery of environments that self-organize relatively-advanced forms of sensorimotor agency whose existence and understanding is fundamental within the enactivist framework for supporting theories about the origins of life and cognition.

In the mechanistic view, one assumes the existence of agents that have well defined physical body and information processing brain allowing them to interact with the rest of the environment through predefined sensors and actuators. Robots for instance are referred as embodied agents: their individuality is clear, as they can easily be distinguished from the rest of the environment, and their self-maintenance is often not a problem, as their body does not change over time except for rare cases of real world or artificially-induced degradation. Hence it is not questioned what makes an agent an agent or even what makes a body a body [238]. Rather, a more central question is to understand how higher-level cognitive processes and sensorimotor adaptivity can arise in the agent through its inter-

1: **individuality:** ability of a self-organizing structure (subpart of the environment) to preserve and propagate some spatiotemporal unity [237], making it a distinguishable coherent entity in the domain in which it exists

2: **self-maintenance:** ability of a self-organizing structure to modify its interactions with the rest of the environment for maintaining its integrity

3: **Autopoietic system:** Introduced by Maturana and Varela [67], the concept of autopoiesis refers to a system capable of producing and maintaining itself by creating its own parts

4: **Organic individuation:** regulation at the metabolic, transcriptional and morphological level to maintain organic integrity [238]

5: active engagement in loops of actions and perceptions in the external environment [238]

6: active engagement in communicative interactions and structural coupling with other agents [238]

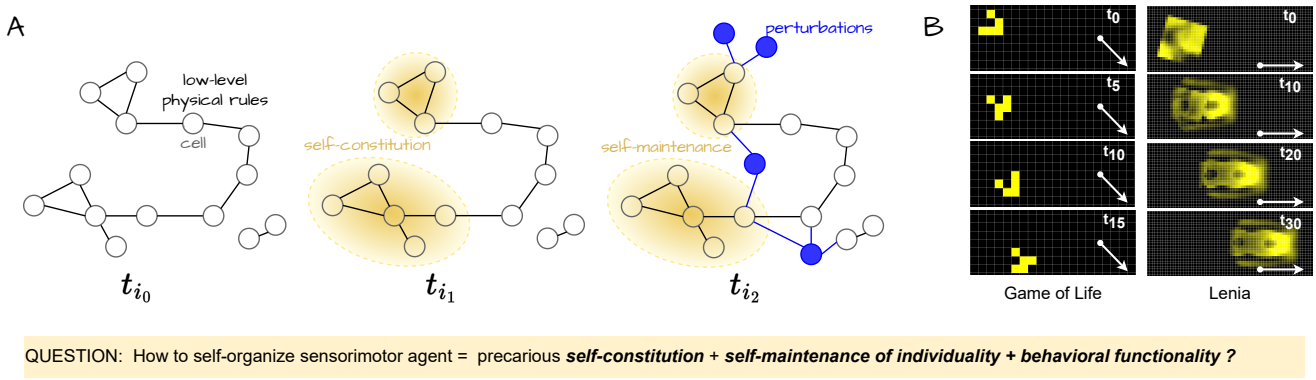


Figure 1.8: **Overview of the scientific question**. (A) The enactivist framework: (t_{i_0}) In the beginning there is only an environment made of low-level elements (cells) and physical laws (local rules). There is no prior notion of agency, no body, no sensor. (t_{i_1}) Agents can come to existence through the coordination of the low-level elements (self-constitution of individuality). (t_{i_2}) To maintain their integrity, agents must sense and react to perturbations using only local update rules (self-maintenance of individuality). (B) In cellular automata models like the Game of Life and a more complex continuous extension called Lenia, it was shown that it is possible to self-organize so-called *gliders* i.e. spatially-localized patterns with directional movement. Directional movement (white arrows) and timesteps are displayed. (Question) In this work, following the enactivist modeling framework, we try to answer the following scientific question: is it possible to find environments in which a subpart could self-organize and be called a "sensorimotor agent"? This would require the existence and emergence of gliders-like structures that not only self-constitute and show motility, but that are also robust to external perturbations and hence must develop some form of sensorimotor apparatus enabling them to make "decision" and "sense" at the macro scale through local interactions only.

actions with the environment. A common methodology is the generation of a distribution of environments (tasks and rewards) and the use of learning approaches, such as deep reinforcement learning, to train the agent's brain to master and generalize those tasks. Within that framework, it was shown that it is possible to engineer agents capable of repertoires of advanced sensorimotor skills such as precise locomotion [197], object manipulation [117], tool use [39] and even capable of adapting the learned behaviors to unseen environmental conditions [44]. Interestingly, they show that the use of curriculum learning[7] is crucial to generate generally capable agents. However, the clear body/brain/environment distinction of the mechanistic framework bears little resemblance with the way information seems to be processed by biological systems. Notably it goes against the concept of morphological computation [240], which argues that all physical processes of the body, not only electrical circuitry in the brain but also morphological growth and body reconfiguration, are integral parts of cognition and can achieve advanced forms of computation.

7: **Curriculum learning:** family of mechanisms that adapt the distribution of training environments to the learner capabilities [198]

The enactive view on embodiment however is rooted in the bottom-up organizational principles of living organisms in the biological world. The modeling framework typically uses tools from dynamical and complex systems theory where an artificial system (the environment) is made of low-level elements of matter (called atoms, molecules or cells) described by their inner states (e.g. energy level) and locally interacting via physics-like rules (flow of matter and energy within the elements) (Fig.1.8-A-t_{i_0}). There is no predefined notion of agent embodiment, instead it is considered that the body of the agent must come to existence through the coordination of the low-level elements (Fig.1.8-A-t_{i_1}) and must operate under environmental perturbations and precarious conditions[8] (Fig.1.8-A-t_{i_2}). Hence, the self-constitution and self-maintenance of individuality are prior conditions for any agency to emerge as it determines the agent's own existence and sur-

8: the idea that bodies are constantly subjected to disruptions and breakdowns [238]

vival [238]. This shifts the problem of "building agents as-they-could-be" to a problem of engineering second-order emergence [241]: how to design environments that can give rise to self-constituting agents that, coupled with the rest of environment, give rise to sensorimotor behaviors? Previous work has shown that the realisation of autopoietic entities in computational media is possible [77, 242–244]. For instance, fully emergent structures showing spatial localization and movement have been discovered, such as the well-known gliders in the game of life up to richer life-like patterns in continuous models of cellular automata (Fig.1.8-B). So far however, two major challenges remain poorly addressed in the enactivist literature. First, autopoietic structures have so far mainly been discovered by human eye and as the result of time-consuming manual search, limiting their discovery and analysis. While some recent works, based on information theory tools, have proposed quantitative measures of individuality in order to facilitate their identification [237, 245], their algorithmic implementation remains difficult in practice. Second, among the very few works that proposed a deeper analysis of the robustness capabilities of the discovered patterns (based on the enumeration of all possible perturbations that a structure can receive from its immediate environment) [78, 244, 246, 247], findings suggest that glider-like structures typically remain quite fragile to external perturbations such as collision with other patterns [246].

In this work, we follow the enactivist framework and consider a class of continuous cellular automata called Lenia [8, 9] as our artificial "world". We show that modern tools from machine learning can help scientists explore the vast space of continuous CA dynamics, enabling to address the problem of engineering robust second-order emergence. We propose a method based on curriculum learning, diversity search and gradient descent, enabling to efficiently shape the search process and to successfully navigate the chaotic outcome landscape of the high-dimensional Lenia system. In particular, we use a family of algorithmic processes called intrinsically-motivated goal exploration processes (IMGEP), an efficient form of diversity search algorithm [248]. While mainly deployed in the fields of developmental robotics [249] and developmental AI to enable robots explore and map vast sensorimotor spaces [250, 251], recent works have shown how IMGEP can also form useful scientific discovery assistants for revealing the range of possible behaviors in unfamiliar systems such as chemical oil-droplet systems [252], physical non-equilibrium systems [253] and models of continuous cellular automata systems as the one considered here [234, 235]. At the difference of these previous works, we introduce two novel elements within the diversity search process: the use of gradient descent for local optimization and the use of stochastic perturbations within a curriculum of increasingly challenging and diverse target properties (hereafter called *goals*). With this method, we are able to find environmental rules leading to the emergence of patterns that self-constitute, self-maintain and move forward under various obstacle configurations, i.e. autopoietic entities displaying robust forms of sensorimotor agency.

We then propose a battery of quantitative and qualitative tests, all formulated within the continuous CA paradigm, to further assess the ro-

bustness and generalization capabilities of the discovered self-organized patterns. Interestingly, the agents also show strong robustness to several out-of-distribution perturbations ranging from perturbing the agent structure in various ways not seen during training (including by a collision with another agent) to changing the scale of the agent. Furthermore, when tested in a multi-entity initialization and despite having been trained alone, not only the agents are able to preserve their individuality but they show forms of coordinated interactions (attractiveness and reproduction), which could be interpreted as a primitive form of intersubjective communication [246]. Those results illustrate the achievable generalization capabilities of artificial self-organizing agents, with respect to their mechanistic counterpart, opening interesting avenues for AI. At the same time, they provide interesting models about the way information might be processed by (brainless) biological agents to ensure robust maintenance of sensorimotor functions despite environmental and body perturbations [254].

1.2.2 Study of sensorimotor agency in continuous CA models

In this work we use Lenia (Sec.1.1.2), a class of continuous CA which is a recently-proposed generalization of Conway's Game of Life [8, 9]. Previous works in Lenia have shown that there exist local update rules f, that can lead to the self-organization of long-term stable complex patterns that display interesting life-like behaviors [8, 9, 235]. Those include forms of individuality (spatially-localized organisation), locomotion (directional movement) and even basic behavioural capabilities (change of direction in response to interaction with other patterns in the grid). However, in previous work, self-maintenance of those behaviors in discovered spatially-localised patterns were typically quite fragile to external perturbations (for example collision with other agents Movie S3), and properties of robustness and generalization were not specifically studied and tested: the possibility to self-organize robust self-maintaining "agents" was still an open question (and this applies to other CAs). Furthermore, these findings have so far relied on handmade exploration, which can be very hard and time-consuming as random rules rarely result in the emergence of localized patterns and even less moving ones (Movie S2).

In this work, we propose to use AI techniques to automate experimentation and the exploration of Lenia, with minimal human intervention. More particularly, the automated experimentation aims to find local update rules f leading to the self-organization of stable (and if possible diverse) agents with sensorimotor capabilities. We also provide tests in order to assess the sensorimotor capabilities of the obtained patterns.

The Lenia environment

We refer to Sec.1.1.2 for details about the Lenia cellular automaton.

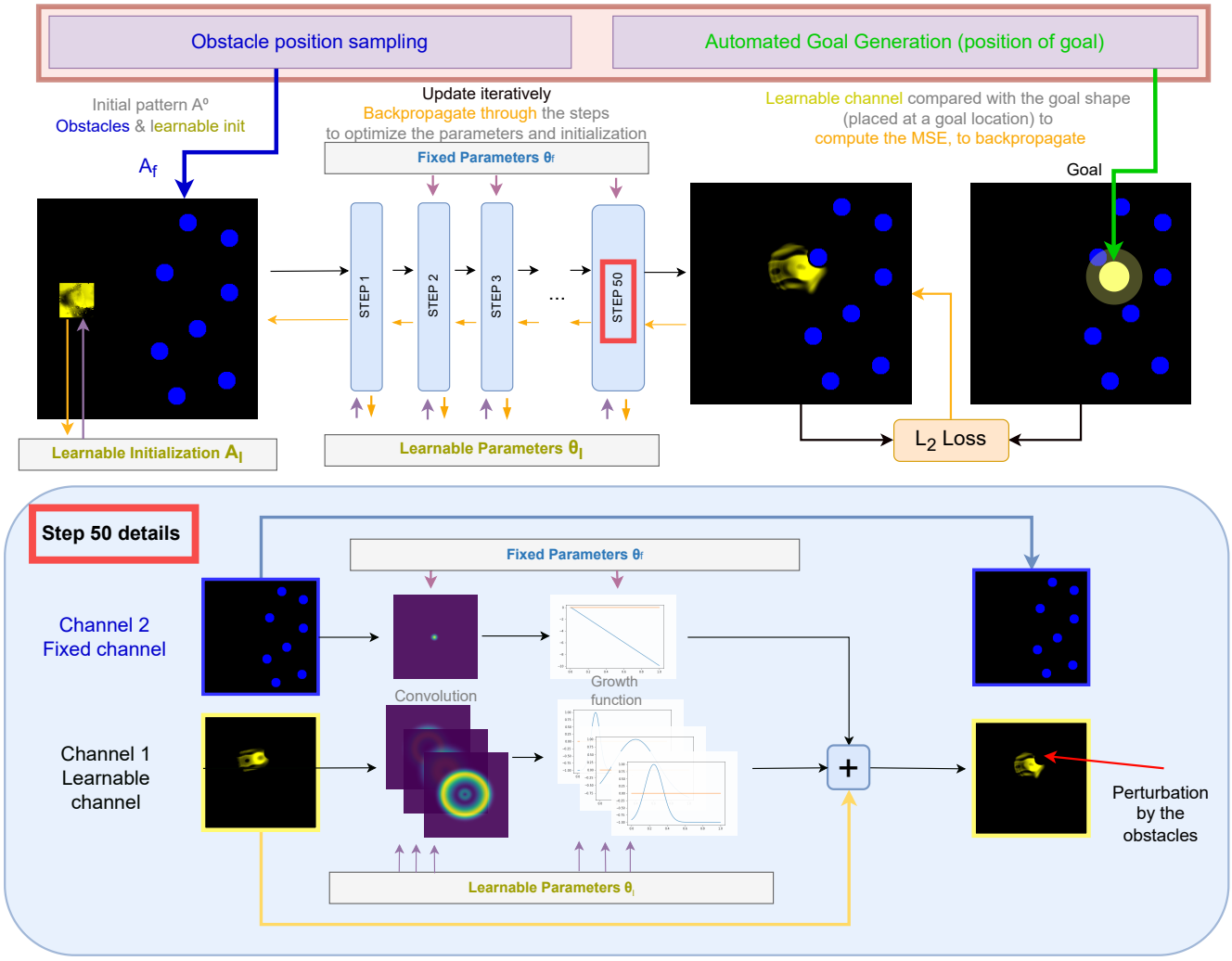


Figure 1.9: **System overview**. (top) Illustration of one experimental rollout with automated (i) generation of target goal (green), (ii) generation of environmental obstacles (blue) and (iii) optimization of learnable parameters toward goal (backpropagation shown in orange). The initial state is iteratively updated by the parameterized rule, we then compute the goal conditionned loss from the last state of the rollout and propagate gradient across the steps to the learnable parameters and initialization. (bottom) Detailed view of a step in Lenia with obstacles. A convolution followed by a growth function is applied on each channel, resulting in a growth update which is added to the current state of the learnable channel. Both the convolution and the non-linear growth function in the learnable channel are parameterized (see appendix A.11).

In this work, we are interested in finding parameters $(\theta, A^{t=1})$ leading to the self organization of moving agents robust to external perturbations from the environment. For this aim, we need to introduce perturbations in the system in a controlled systematic way, both for testing the robustness and as criteria during the search. However, due to the dynamical nature of the system, controlled perturbations over several steps in the CA system are often hard to introduce. To help solve this issue, we propose to take advantage of the multi channel version of Lenia (see Sec.1.1.2) and separate the low level elements of the system in two types: the first "fixed" channel, which is hand-engineered, introduce elements that act as stable controlled obstacles (blue in Fig 2.); the second "learnable" channel, where parameters of the physic are learned, is where the agent has to emerge (yellow in Fig 2.). In practice, the environment parameters $(\theta, A^{t=1})$ are then separated in two. The first part, denoted $(\theta_f, A_f^{t=1})$ is a hand engineered part where θ_f gives the rule on how obstacles block matter from going in, while

$A_f^{t=1}$ gives the obstacle placement and shape. Details on how we implement obstacles as part of the CA rule can be found in material and methods (Sec.1.2.4). The second part however, denoted $(\theta_l, A_l^{t=1})$, is free: the method presented below enables to learn these environment parameters so that "agents" with sensorimotor capabilities can self-organize.

What we are searching for is thus learnable parameters $(\theta_l, A_l^{t=1})$ that will induce a physic leading to the self-organization of agents that are able to move and survive in a grid where obstacles perturb their structure and therefore may break their integrity. Note that finding pattern with such capabilities is not trivial, for example moving patterns found by hand in [8, 9] (as the Lenia glider), which are stable without perturbations, often die from the collision with our engineered obstacles (Movie S4). Note that in our system, if an agent is to emerge, the only way it can "sense" previously-introduced obstacles is from the perturbations that the obstacles induce on its structure. Compared to the physical world, the agent does not "sense" the obstacles by means of exchange of particles like photons or chemical molecules, as in vision or chemoreception, but more akin to direct touch as in haptic perception.

Intrinsically Motivated Goal Exploration Process (IMGEP)

Formally, a set of parameters $(A_f, \theta_f, A_l, \theta_l)$ in Lenia maps to a certain sequence of states (trajectory o). This trajectory can then be mapped to a vector $R(o)$, through a defined characterization function R. This vector provides a behavioral description of the trajectory, and the image of R represents the space of possible behaviors that can emerge in the system. As we will show below, randomly exploring the space of learnable parameters (A_l, θ_l) is both costly in terms of experimentations, and inefficient for finding robust sensorimotor behaviour.

Thus, we propose to leverage an AI technique called Intrinsically Motivated Goal Exploration Process (IMGEP) [249] to help exploring the space of behaviours. As this technique was originally developed to model curiosity-driven exploration in children [255], we call such a system a *curious automated discovery assistant*. The IMGEP technique relies on *goal-directed* search, which we leverage to drive the system toward the emergence of diverse target (sensorimotor) behaviors, called goals. More precisely, given a goal-sampling strategy G, IMGEP automatically samples target *goals* $g \sim G$ which are points in the behavioral space. For each goal g, the objective is then to optimize toward parameters (θ_l, A_l) leading to a sequence of state which is mapped as closely as possible to this goal. To score the trajectory according to a goal, a loss function $\mathscr{L}(g, o)$ taking as input the trajectory and the goal is used.

The behavioral descriptor R we choose in this contribution is the position of the center of mass at the last timestep of a simulation. The behavioral space then consists of all possible (x,y) coordinates in the grid. The objective for a given goal $g = (x, y)$ is thus to find parameters (A_f, θ_f) leading to the emergence of a spatially localized

pattern attaining the goal position at the last timestep under several perturbation by obstacles. In this work, we choose to define the (goal-conditionned) loss as the mean squared error (MSE) between the state at the last timestep of the trajectory and a disk centered at the goal position. In addition to closeness to the goal position, the loss function we use incentivizes localization of the mass to prevent pattern explosion and collapse, which is a very common outcome of Lenia parameters. We then use gradient descent to optimize the learnable parameters $(\theta_l, A_l^{t=1})$ by backpropagating the loss through the steps and make progress toward the goal (Fig 2.).

Gradient descent optimization has already been successfully applied with cellular automata [256] on learning CA parameters leading to the growth (and regrowth) of a target pattern [257] or texture [258], or enabling cellular collectives to perceive their large scale structure [259], proving the effectiveness of such method (with some additional component for training for long term stability) in complex chaotic self-organizing dynamic. However, in this work, we consider moving agents which are a fragile type of pattern in Lenia as moving forward in such system means to grow new cells at the front while the ones at the back die. This equilibrium between growth and death is also challenged by the random perturbations we introduce in the system. This means that changes of parameters, because of the chaotic nature of the system, can easily break the equilibrium between growth and death of cells making the optimization harder.

To help with this difficult optimization landscape we propose to introduce a *curriculum* for making small improvements iteratively. Curriculum learning has already been applied for optimizing cellular automata rule with gradient descent as a solution for getting out of a trivial local optima in Variengien et al (2021) [53]. The curriculum also solves technical gradient flowing problem, detailed in appendix 11.

The intuitive idea behind our curriclum is to first learn rules leading to moving (spatially localized) agents which we train to go further and further (in the same amount of timesteps, hence faster) and at some point train them to go further while dealing with obstacles. To do so, the fixed environment $A_f^{t=1}$ we sample for training has a certain structure: the left half of the grid is free from obstacles while the right part contains obstacles that will be randomly placed at every rollout (blue in Fig.1.10-a). The sampling strategy G we chose in the IMGEP also participates in the curriculum as it is biased to randomly sample goals that are a little bit further than previously attained positions. More information on the sampling strategy can be found in appendix 7. Putting target goals in the obstacle area means that during training, the potentially emerging agents will have to go to a specific location while its structure is perturbed by obstacles randomly placed. The gradient descent optimization will incentivize recovery from perturbation and to keep moving despite being damaged. In addition, the fact that the obstacles are randomly placed should incentivize generalization to different perturbations.

To sum up, the IMGEP iteratively (and automatically) generates increasingly difficult goals, in increasingly difficult and diverse environments, for which we will try to find, and optimize using gradient descent, learnable parameters $(\theta_l, A_l^{t=1})$ that will lead to the self organi-

zation of agents achieving these goals. For each goal (position), the optimization steps are done under several obstacle configurations $\{A_f\}$ in order to learn to resist to different perturbations. After each optimization, we then test the final obtained parameters on several obstacles configurations $\{A_f\}$, that are sampled the same way as in the training steps, to assess the reached position. We store this (parameters, reached position) couple in history $\mathscr{H}$ in order to be able to use it as a starting point for subsequent goals. A more detailed description of the method can be found in material and methods (Sec.1.2.4).

Evaluation of the discovered patterns

Whereas the notion of *agency* is closely tied to the ability of an organism to maintain its own organization despite encountering novel circumstances, the robustness of current artificial autopoietic systems is lagging far behind the robustness of their biological counterparts. We believe that this limitation, together with the difficulty of engineering such autopoietic systems, is a major reason why we have not assisted yet to a wider adoption of the enactivist framework by the AI community. The IMGEP search, which is precisely intended to facilitate the search of such autopoietic systems, should provide us with a database of parameters $\{(A_f, \theta_f)\} \in \mathscr{H}$ that (when successful) lead to the self-organization of patterns that are robust (at least) to the different obstacle configurations seen during training.

To go further and characterize *agency* and the degree of *robustness* of the discovered parameters/patterns, we propose an empirical evaluation procedure in two stages. First an "empirical agency filter" is used on the database of discoveries to discard parameters that do not lead to the self-organization of what we call "agents" in Lenia. More precisely, our filter implements several classifiers, inspired from ones proposed by Reinke et al. [234], to detect whether the emergent matter does not disintegrate (vanishes or explodes), forms a coherent entity (single soliton), and does so during a long-enough time window (longer than training). In addition to the agency filter, we also introduce a moving filter which tells if an agent is moving (travels a minimum distance) or not (examples of discovered "agents" that are considered not moving are shown in Movie S19). Then, to assess the capabilities of selected agents to withstand perturbation by obstacles we perform a basic obstacle test: testing them on obstacle configurations similar to the ones seen during training; and various generalization tests: running them through a battery of tests with several *out-of-distribution* perturbations that were not seen during training. In particular, we test the discovered sensorimotor agents to harder obstacle configurations, stochastic cell updates, changes of initialisation and changes of scale that were not experienced during training. For each test, given a distribution of perturbations, we measure *robustness* as the average performance over sampled perturbations, where performance is a binary success metric that determines whether the agent "survived" the perturbation or not. As for "survival" metric, we simply apply our agency filter to detect whether the (perturbed) emergent entity is able to self-maintain despite the

introduced perturbations (i.e. is still an agent at the end of the test). Note that this metric closely follows the definition of *cognitive domain* of an autopoietic system, which was introduced by Maturana and Varela [67] and later defined by R. Beer as the percentage of *non-destructive*[9] perturbations, out of all possible perturbations, that the autopoietic system can tolerate [78]. Because measuring the cognitive domain as such would require an exhaustive enumeration of all possible perturbations and all possible valid states that the entity can take, which is not tractable in the Lenia environment, we instead rely on a proxy metric and on a set of chosen empirical tests. Finally, in addition to robustness, we also measure the performance of agents in term of *speed* with and without obstacles, especially as speed can be a measure of performance of motor capabilities (for example for biological agent to flee predators or chase preys) and as speed with obstacles is an interesting measure on how well the agent deals with obstacles. We refer the reader to Material and Methods (Sec.1.2.4) and to appendix A.1.8 for more details on our evaluation procedure.

9: a perturbation is said to be *destructive* if it fundamentally disrupts the entity's organization leading to its disintegration [78]

In addition, we provide the code[10] enabling to reproduce all results, as well as an interactive web-demo[11] where one can replay the discovered agents and test them to all sorts of freely-drawn perturbations including custom obstacle shapes, addition and/or removal of mass, interactions with other agents in the grid and control of environmental cues (attractive elements) in the Lenia grid.

10: Source code for reproducing the results can be found at `https://github.com/flowersteam/sensorimotor-lenia-search`.

11: Interactive web demo and additional videos can be found at `http://developmentalsystems.org/sensorimotor-lenia-companion/`

We argue that those quantititative and qualitative tests, which were all implemented within the continuous CA paradigm, can serve as a good baseline to evaluate the generalization capabilities (and hence the degree of agency) of autopoietic systems in enactivist research, akin to commonly deployed benchmarks in AI for evaluating mechanistic forms of agency [44].

1.2.3 Results

In this subsection, we analyze the discoveries made by the proposed approach (IMGEP) and compare it with two other exploration baselines: a *random search*, where parameters are sampled uniformly in the parameter space (same ranges than for the IMGEP, given in appendix A.1.6); and a *handmade search*, where we collected the discoveries, made by semi-automatic search and expert selection, presented in the original Lenia papers [8, 9]. Each IMGEP experiment outputs 160 parameters but performs in average 11700 Lenia rollouts, due to stochasticity in the method (see Materials and Methods). For IMGEP and random search, 10 independent repetitions are performed (where random search is given the same experimental budget of 11700 rollouts per seed). Note that the comparison with handmade search, while interesting, is challenging in practice as it is the result of tedious search for which the total experimental budget is unknown, and which was conducted over some Lenia hyper-parameters that are not all included in the automated search (e.g. various number of channels or kernels). Moreover, we use a slightly different parameterization of the rule to allow for differentiability (details in appendix A.1.6).

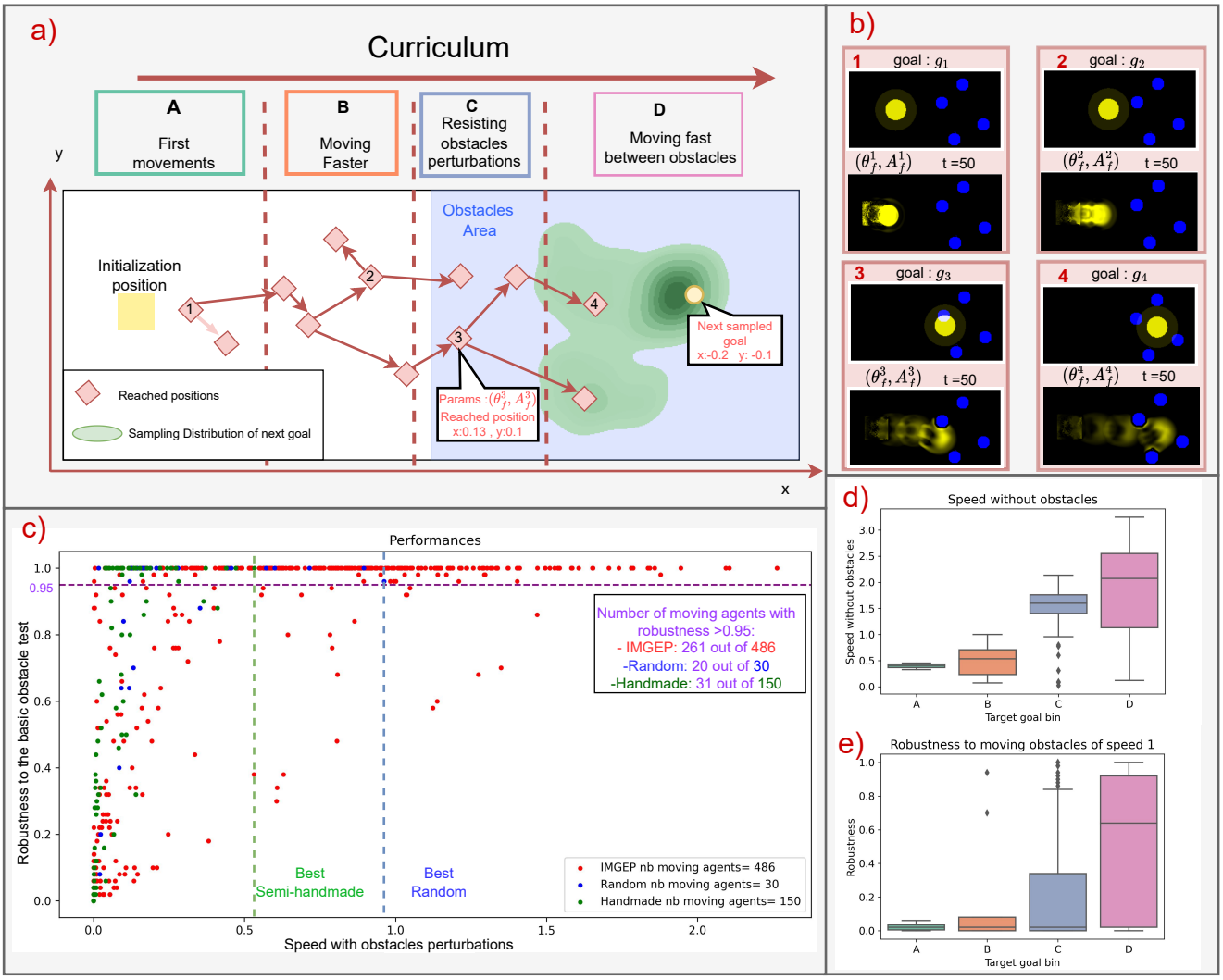


Figure 1.10: **Curriculum and performances**. a) Schematic view of the curriculum. The curriculum iteratively sample goal positions (yellow disk), further in the grid, starting from very close to the initialization (A) to further away without obstacles (B) to further away in the obstacle area (C, then D). Arrow between reached positions (red square) represent that the parameters leading to a pattern attaining the tip of the arrow position was initialized before training by the parameters reaching the back of the arrow position. b) Examples of patterns obtained along the curriculum as well as their associated goal. We observe patterns going further and further in the same amount of steps (50 steps) and for the latter dealing with obstacles in their way. To display the trajectory of the agent in the learnable channel (yellow) we superposed the frames over all timesteps putting more transparency in earlier timesteps. c) Performances in term of robustness to the basic obstacle test and speed with obstacle perturbations of the moving agent produced by: IMGEP (red), random parameters search with the same computation as our method, i.e. 117 000 parameters tried in total (blue) and handmade agents found in the original Lenia papers (green). d,e) Distribution of the Speed without obstacles perturbation (d) and robustness to moving obstacles (e) of moving agents obtained by the IMGEP along the curriculum. Details on these metrics can be found in Appendix.A.1.8,A.1.8. We observe that the curriculum is translated in an improvement in the 2 presented quantities.

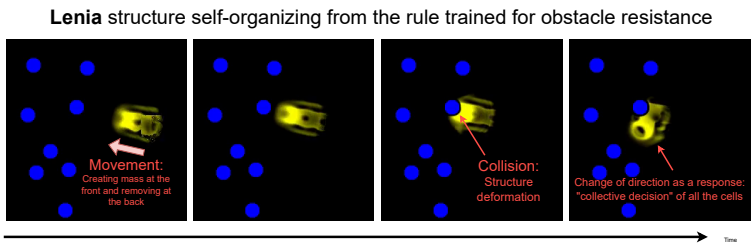


Figure 1.11: Robust moving structure emerging from the rule of the environment that are discovered by our method

For the three baselines (IMGEP, random search and handmade search), we filter the obtained parameters to select only the moving agents (passing the agency and moving test) and measure their speed and robustness to the basic obstacle test and generalization tests, as described in the previous subsection.

Individuality, locomotion and sensorimotor capabilities

As illustrated in Fig.1.10., the IMGEP search enables to evolve agents along a curriculum which progressively leads to the emergence of individuality, locomotion and sensorimotor capabilities. At first, the IMGEP samples goals (i.e. target positions) that are not too far from initialization (area A in Fig.1.10-a) and enabling to find rules leading to the self organization of spatially localized patterns which starts to move a little bit from initialization (as shown in Fig.1.10-b-1). Then, from these newly learned rules the IMGEP samples further goals (area B in Fig.1.10-a) which lead to spatially localized patterns that move further in the grid in the same amount of time (Fig.1.10-b-2). At this point, some obtained parameters already lead to the self-organization of *moving agents* i.e. passing our empirical agency test and moving tests (long-term stable solitons capable of moving while self-maintaining). Moving agents' patterns are in fact already not trivial to find through random search in the parameter space as only 30 moving agents were found through the 10 seeds of random search out of a total of 117 000 trials of parameters. The speed of the obtained moving agent at this point is still limited as can be seen in Fig 3-d.

The IMGEP pursues the curriculum, taking advantage of the previously learned parameters that already result in moving agents, now sampling target goals that are even further away from the initial position, in the obstacle area C,D in 3.a, leading to moving agents entering the obstacle area (as shown in Fig.1.10-b-(3,4)). As expected, the parameters resulting from those goals have a higher robustness to obstacles as can be seen in Fig.1.10-e. We refer to appendix A.1.3 for extra experiments with an ablation of the obstacle area during optimization showing that the increase of robustness is due to the presence of obstacles in the optimization and not only to the distance of the target goal position to the initialization.

As expected, we observe that agents trained with further goals move on average at faster speeds in environments without obstacles (Fig.1.10-d)

At the end of the curriculum loop, the obtained rules often lead to the self-organization of moving agents that are able to navigate fast in an area with obstacles while still maintaining their integrity (Fig.1.10-b-4, Movie S1). The emerging agents are capable of changing direction and recovering in response to perturbations induced by the obstacles, i.e. have sensorimotor capabilities, and this only through the global coordination of those identical low-level parts and in particular without having any central unit computing decisions.

In total, 9 out of the 10 seeds led to at least one sensorimotor agent, which we define in this contribution as a moving agent with a measured robustness ≥ 0.95 in our basic obstacle test. Note, however,

that the performance in terms of speed with obstacles varies from one seed to another (see Appendix Table A.1).

Over the 10 seeds, a great part of the obtained emerging moving agents are sensorimotor agents. In fact, over 10 seeds, 486 of the 1600 parameters (10 seeds x160 parameters) led to moving agents according to our empirical agency and moving filter, from which 261 have a robustness to obstacles ≥ 0.95.

As a comparison, out of the 117 000 parameters generated by the 10 seeds of random search, only 30 led to moving agents from which 20 have a robustness to obstacles > 0.95. Our method surpasses random search in term of speed with obstacles and robustness of the obtained agents, as well as the total number of long term stable moving agents obtained as can be seen in Fig.1.10-c (486 for IMGEP and 30 for random search in total over 10 seeds and with the same Lenia rollout budget). In fact random search is able to find some agents ($\sim 1\%$ of all its discoveries) but most of them are static compared to IMGEP whose directed search fosters the emergence of moving agents (Movie S5).

Our method also results in agents with better robustness and speed than the ones found in the original Lenia papers [8, 9] (Fig.1.10-c).

Ablation studies of the method can be found in the appendix A.1.3, showing how curriculum, diversity search and gradient descent are key ingredients in the method and are an efficient direction to search for sensorimotor behavior in self-organizing systems. We also provide the sequence of reached positions of a seed in appendix A.1.2, displaying the curriculum and showing how diversity search can help find potential stepping stones.

Generalization

Biological organisms are able to maintain phenotypic stability in the face of diverse environmental perturbations arising from external stresses, intracellular noise, and even quite drastic changes during morphogenesis such as perturbations to the embryo structure [260] or to the substrate cellular size [261]. It has long been recognized that robustness is an inherent property of all biological systems that has been strongly favored by evolution [262]. In this subsection, we are interested to see if similar robustness capabilities can be achieved by the artificial self-organizing agents that have been discovered by our artificial evolution workflow (Fig.1.12 and Fig.1.13). To do so we evaluate the generalization capabilities, over the proposed battery of tests, of the 10 best agents discovered by the IMGEP, random and handmade search variants, as well as on the agents that have a speed within obstacles greater than one (91, all discovered by IMGEP). "Best" here is computed according to the speed-robustness criteria presented in Figure 3-c, i.e. the fastest with obstacle that also have a robustness in the basic obstacle test> 0.95. The performances are fully reported and compared in appendix tab.A.2. As we will see, the discovered agents showcase quite impressive generalization capabilities at the organic, sensorimotor and inter-subjective levels [238]. We group the

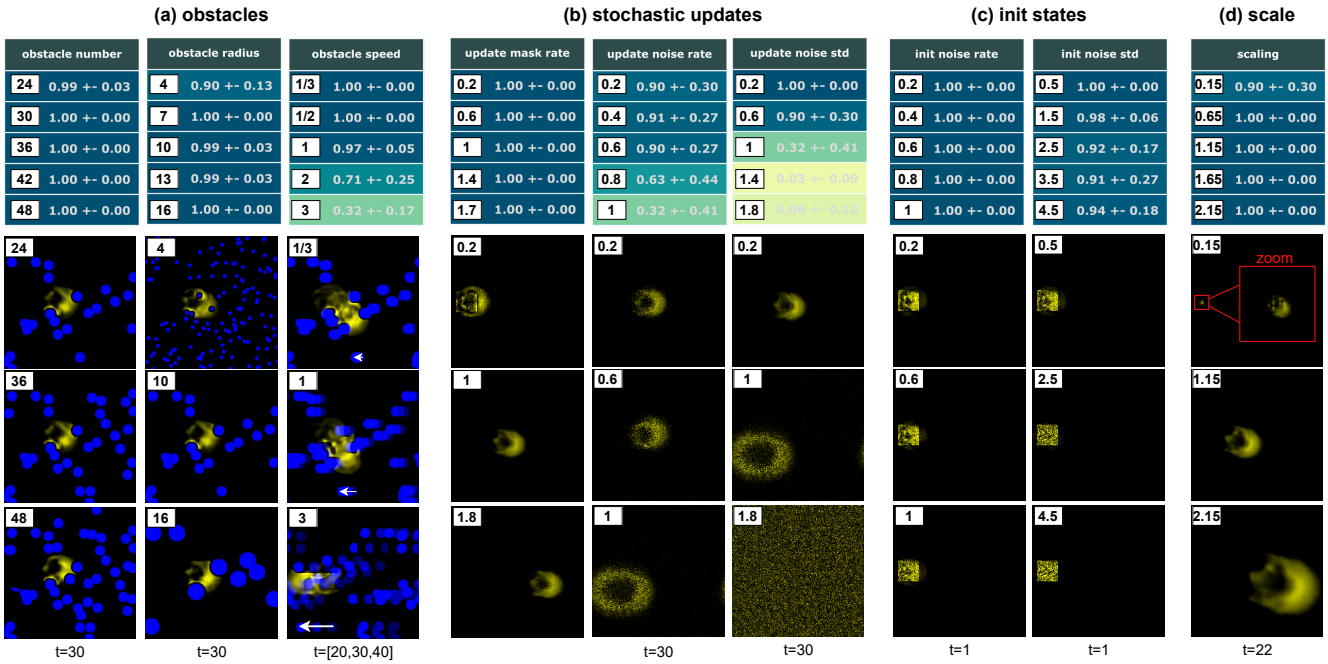


Figure 1.12: **Quantitative tests of generalization of the discovered sensorimotor agents**. We conduct a battery of quantitative tests which we organize in 9 families of parameterized perturbations that test for various (a) obstacle number, size and speed, (b) rate of cell updates, as well as rate and magnitude of noise added to the updates, but also (c) rate and magnitude of noise added to the initial state and (d) scaling factors. For each family, we test for 5 different parameter values, i.e. perturbation strength, resulting in a total of $9 \times 5 = 45$ tests. For each test, the performance of an agent is computed as the average score of survival over 10 random seeds. A score of 1 (dark blue) means that the agent survived all 10 tests whereas a score of 0 (light yellow) means that the agent survived none of the tests. The table reports the mean and standard-deviation performances, over the 10 best agents discovered by our goal-directed curriculum, for all of the 45 tests (one table cell per test), where "best" is determined by the speed/robustness criteria introduced in Figure 3-c. Below each column, we show snapshots of system rollout at test time given the newly introduced perturbations. The shown snapshots are all taken from rollouts of the "best" agents, and from the first seed (out of the 10 tested random seeds). Timesteps are specified under the images, for instance snapshots of the perturbations applied on the initial state are shown at t=1.

observed generalization capabilities into six categories: harder obstacle configurations (external stresses), stochastic cell updates (per-cell noise), changes of initialization ("embryo" variation), changes of scale (compute capacity variation), interactions with other agents in the grid (inter-agents regulation) as well as with human-controlled environmental cues (observer-agent regulation).

Harder obstacles. We first tested the agents generalization capabilities to a larger and more challenging set of obstacle configurations. The test set includes controlled configuration with varying number, size and speed of obstacles (Fig.1.12-a), as well as human-drawn obstacles such as vertical walls and dead ends (Fig.1.13-a). Interestingly, whereas some well-placed perturbations can lead to death or explosion, the discovered agents show strong robustness and generalization to most of the test set configurations. They showed quasi-perfect survival to grids with up to 48 obstacles, to grids with small (but dense) or big (but sparser) obstacles, and to obstacles with moderate speed. High-speed obstacles however, seem to challenge agent's survival (Fig.1.12-a), even though the IMGEP-discovered agents are still much more robust to moving obstacles than the ones discovered by random and handmade search (appendix Table A.2 and Movie S6). Those results suggest that, by training for fast-moving and obstacle-resisting behaviors, our goal-directed curriculum favored the self-organization

of agents that are able to *quickly* recover from perturbations induced by the environment, even ones not seen during training. For instance, qualitative tests also showed that the discovered agents are able to successfully navigate forward while coming across tightly-packed obstacles, walls of various inclinations, corners, dead ends and even bullet-like types of obstacles (Movie S10).

Stochastic updates. We then tested the agents generalization capabilities to asynchronous and noisy cell updates. As proposed in Mordvinstev et al. [257], relaxing the traditional assumption of synchronous update in cellular automata (which assumes a global clock) is closer to what you would expect from a self-organized system, and can be done by applying a random update mask on each cell (parameterized by the update mask rate in Fig.1.12-b). Despite the update mask enforcing asynchronous and less (or more) frequent cell updates at test time, the discovered parameters still give rise to self-organized agents that perfectly self-maintain (survival scores of one) and that showcase very similar morphology and behavior as the agents with synchronous updates (Movie S14). The agents are slowed (or fasten) a little bit but this is what we can expect as each cell is updated in average only a fraction of the time (or several times per timestep). We also relax the assumption of exact update by adding random noise, of various amount and magnitude, to the cell states during the system rollout. Here, we observe that the agents can resist quite consequent quantities of noise but passed a certain level, as expected, the collective looses its integrity and disintegrates (Fig.1.12-b).

Changes of initialization. While the initialization pattern has been learned with a lot of degree of freedom (pattern in $[0,1]^{40\times40}$), we can look if similar patterns (phenotypes) can self-organize from other (maybe simpler) initialization patterns. This capacity to converge to the desired anatomy in spite of a different initialization ("embryo"), is something that can be found in biological organisms [260], and that we can expect in our system as well. Indeed, as shown in Fig.1.12-c, we can see a quasi-perfect robustness to noise-altered initial states, and this even for quite high amounts of noise (except for few configurations that lead to death). These results suggest that the final phenotype forms a strong attractor towards which the different initial mass pattern tend to converge under the learned CA rule. The learned CA rules are hence prone to encode, grow and maintain a specific target morphology (and its associated functionality), which is consistent with the agent ability to recover from obstacle-induced perturbed morphology. As illustrated in Fig.1.13-b, we also tested for handmade initial patterns such as bigger disks and same-size asymmetrical disks (for example with gradient activation). Interestingly the large disk initialization led to multiple entities forming and separating from each other. The same-size disk, which is much simpler than the trained initial states (but preserves some form of asymmetry) also converged toward the same morphology. However the robustness to initialization is not perfect as many initializations, such as disk of smaller size and/or without asymmetry, easily lead to death (Movie S18).

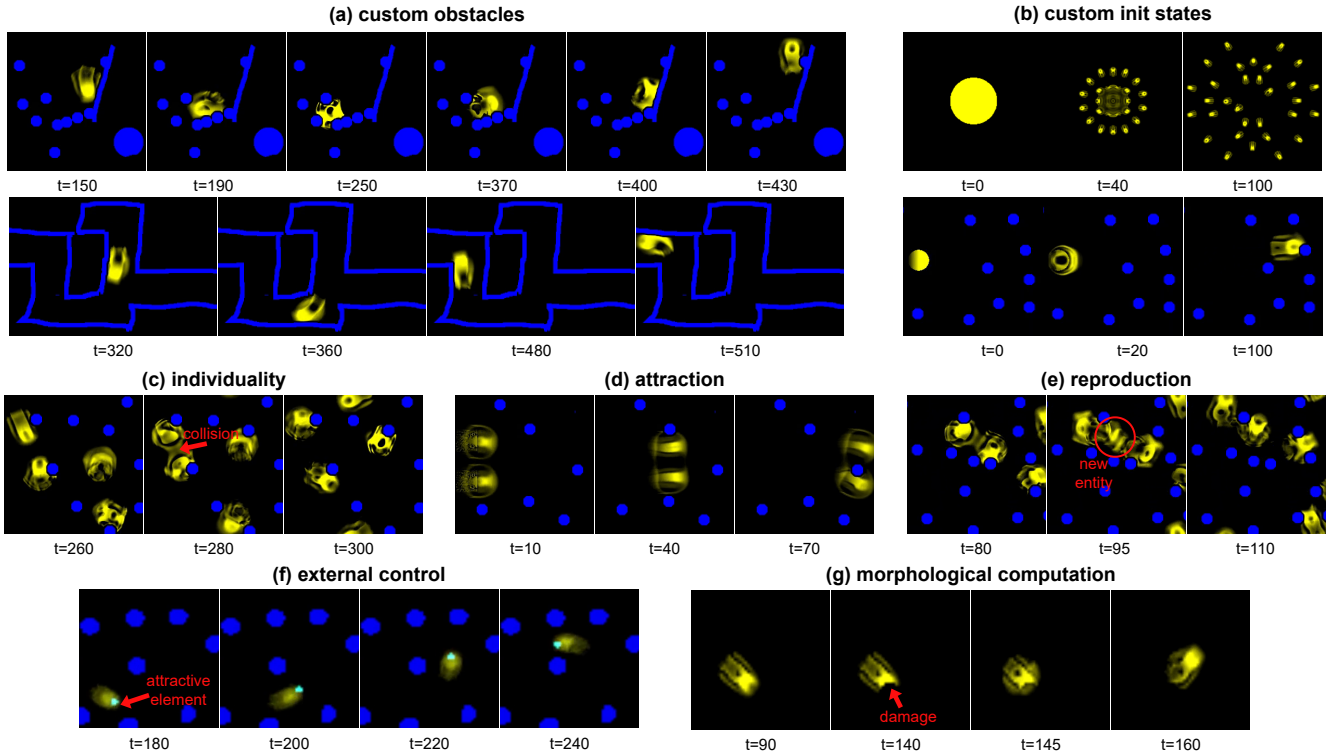


Figure 1.13: **Qualitative tests of generalization of the discovered sensorimotor agents**. We conduct a battery of qualitative tests, where we test the (best) discovered agents to all sorts of difficult perturbations including (a) freely-drawn obstacles such as walls, mazes or dead-ends (b) freely-drawn initial states such as very big disks (resulting in the emergence of multiple entities) or small disks with gradient asymmetry, (c-d-e) introduction of other agents in the grid (resulting in the emergence of inter-agent interactions such as individuality maintenance, attraction and reproduction), (f) the introduction of novel low-level elements that have an "attractive" effect on the agents (allowing external user to guide the agent trajectory in the grid); and (g) custom mass removal (pixel erasing). Details of the resulting observed behaviors are provided in the text, with videos available on the companion website *https://developmentalsystems.org/sensorimotor-lenia-companion/*.

Changes of scale. Similarly, while the initialization and update parameters have been learned at a certain spatial resolution during training resulting in agents of a certain size (in term of number of cells), we can artificially change the scale at test time by approximate resizing of parameters (see Appendix subsection A.1.8). As shown in Fig.1.12-d, we tested for different down-scaling (and up-scaling) factors that surprisingly resulted for most of them in fully functional agents with the overall same structure but smaller (or larger) size in terms of number of cells. For agents which are down-scaled, and hence have much less pixels/cells to do the computation, it is particularly surprising that they are still able to sense and react to their environment and still show relatively-advanced levels of robustness (Movie S15). This scale reduction has a limit (a scaling of 0.15 already leads to some death) but we can go quite far down and still obtain functional phenotypes. For the bigger agents, which therefore have more space to compute (but also more cells to organize), we observe similar results where agents still self-organize to functional phenotype. Once again, this resonates with findings in biology suggesting that organisms are able to accommodate cell-size differences by adjusting cell number in order to maintain roughly constant body size and structure [261].

Interactions. We were then interested to test how the discovered agents would react when interacting with other agents in the grid. Given the set of parameters (A_l, θ_l), we can trigger the forming of several macro-entities at test time by replicating the initialization square pattern $(A_l \in [0,1]^{40\times40})$ at different locations within a larger grid $(A^{t=1} \in [0,1]^{512\times256})$ and letting the system unroll. Doing so leads to the development of several entities of the same "specie" (governed by the same update rule/physic θ_l). As illustrated in Figure 5, we did that for several of the discovered sensorimotor agents, and qualitatively observed several interesting emergent interactions.

The first thing that we observed is that, several of the discovered agents show strong *individuality* preservation (Movie S11). The fact that the individual agents do not merge nor enter in destructive interactions despite being all made from identical cells is an intriguing example of how the boundary of a "self" [263] can emerge and maintain in self-organizing systems. In particular results suggest that, in the Lenia system, individuality can be obtained as a byproduct of training an agent alone. Our intuition is that by trying to prevent too much growth during training, it learned to prevent any living cell that would make it "too big", including living cells from other entities here.

A second type of interaction that can be observed with certain parameters/environments is *attraction*. As illustrated in movie S13, two agents placed in the same grid can show attraction when coming close enough from one another, leading them to stay together and move in the same direction. Interestingly, when they encounter an obstacle, they are able to separate briefly and then to reassemble together. Similarly, even when they stay together, we can still qualitatively observe two distinct entities that are interacting with one another while maintaining their overall shape and integrity. This type of behavior has been studied in the game of life under the concept of *consensual domain* [246].

A third type interaction that has been observed in some of the discovered agents is a form of *reproduction* where collision between two agents give rise to the birth of a third entity (Movie S12). This kind of interaction seems to happen when one of the two colliding entities is in a certain "mode", like when it just hit a wall. Our intuition is that when it hits a wall, the self-organizing agent produces a growth response in order to recover. During this growth response if there is extra additional mass coming from another entity then the self-organizing agent might split off from the created mass while the separated mass, from robust self-organization (see "Changes of initialization" above), grows into a complete individual.

External control. A central challenge in synthetic biology, when faced with unconventional forms of agency such as collective of cells, is to find new ways to communicate with the cells to induce desired behaviors at the collective level without having to physically "rewire" the structure of the agent (e.g. via genome editing) but rather by introducing externally-controlled cues in the environment [264]. Here, we are interested to see whether we can induce (novel) target behaviors in

the discovered agents without having to modify the learned parameters θ_l. In particular, we investigate whether the agents can show *attraction* to some novel elements in their environment (like in nature organisms being attracted to certain chemicals, lights or temperatures) and if we could use those elements to guide the macro-entity. To do so, we introduce a new type of "attractive" low-level elements within the Lenia CA paradigm. More precisely, given the set of learned parameters θ_l, we introduce a novel local rule with parameters θ_a that determine the physical influence of the attractive elements onto the agent cells. To find parameters θ_a triggering the desired attraction effect at the agent behavioral level, a simple random search with automatic pre-filtering and final human assessment was performed (see appendix A.1.8 for details on the procedure). Movie S17 is an example of obtained behavior where we can clearly see that the sensorimotor agent is getting attracted to the newly-introduced environmental element (disk of cyan particles) which allows the external user to "control" the agent trajectory by moving the disk in the grid. Interestingly, in spite of this novel behavior, agents are capable to maintain their normal sensorimotor capabilities showing robustness to collision with obstacles and other agents in the grid. Besides, once the attractive element is removed the agents return to their normal behavior. However adding extra rules also fragilize equilibrium that existed in the agent rules as it creates perturbations that the agent has not been trained to withstand, leading sometimes to death or explosion (or to other behaviors such as reproduction due to extra boost of growth). Once again parallels can be drawn with findings in biological organisms, for instance [265] show that controlled UV light beam can be used to externally guide the trajectory of micro-swimmers to perform on-demand drug discovery. While we only tested for attraction-type of generalization behaviors, we believe that more sophisticated types of environmental guidance could be induced, though probably necessitating more advanced search methods.

Morphological computation. This subsection has provided several empirical evidences of how adaptive high-level functionality can emerge from a collective of low-level, decentralized elements. In order to withstand the tested perturbations, the cellular collective first needed to "sense" the induced perturbations through a deformation of the macro structure. After this deformation it had to "communicate" the information and make a collective "decision" on where to grow next. Then it had to move and regrow its shape, altogether giving rise to the observed robustness of the macro structure. In order to better visualize the physical manifestation of decision-making within the cellular collective, we manually suppressed a part of the agent (Fig.1.13-g, Movie S16). We can clearly observe that perturbation of the macro-structure is what leads to the direct change of direction. Those results support the fact that computation of the decision is made at the morphological level hence that morphology, decision-making and motricity are highly entangled phenomena [240].

1.2.4 Materials and Methods

System. An update in Lenia is given by the different rules composing the function f_θ, each rule is composed of a convolution kernel (which will sense the surrounding of the cell) and a growth function (a function which will convert this sensing, a scalar, into an update of the mass, another scalar). The update of the cells are then given by a weighted sum of the update given by each rule. At each step, the calculation of the update is done identically on every cell of the grid (every cell apply the same convolution filter and growth function). This update is then added on the associated cell and the result is clipped between 0 and 1. See figure 2. for an illustration of the update. The Lenia system used in this work is slightly different from the one in the original paper [8, 9]. We changed the parameterization in order to allow more gradient to flow through the steps (more details in appendix A.1.6). We also choose to use 10 rules, from the learnable channel to itself. We refer to appendix A.1.6 for a detail on the parameter of the systems and their role. In total the 10 rules are controlled by 132 parameters.

Modeling of Environmental Constraints. The parameter θ_f gives the update rule associated with obstacles. This rule senses in the obstacle channel and update in the learnable channel. This means that the convolution will be calculated upon the obstacle channel and the growth obtained through the growth function will be added to the learnable channel. In practice, for θ_f, we use a rule with a convolutional kernel of small size, so that obstacles have effects only locally and a growth function which has a huge negative decrease of mass in the learnable channel to prevent any matter from going where obstacles are present. More information in appendix A.1.6.

IMGEP. Our proposed method, based on the IMGEP framework [249], and fully described in appendix A.1.7, starts by initializing the history with 40 random parameters and their associated reached position (position of the center of mass at last timestep) computed over 20 rollouts with random obstacle configurations. The method then begins a loop where each step is composed of 1) the sampling of a goal (x,y position in the grid) , then 2) a selection from the history of the parameters reaching the closest goal which will be used to initialize the parameters, 3) an optimization of those paramaters towards the goal under several obstacle configurations, 4) a test of those parameters over 20 obstacle configurations to compute the final reached position after optimization, and adding the couple (parameters, reached position) to the history to reuse it in next steps. Pseudo code 2 and figure A.12 illustrating the IMGEP algorithm can be found in appendix. Details of each step of the method: 1, 2, 3, 4 can be respectively found in appendix 7, 11, 11, 11.

In this work, the loop defined above is composed of 120 outer steps where 1 out of 5 outer steps performs 125 steps of gradient descent

while the rest performs random mutation on the initialized parameters and 15 steps of gradient descent (details on mutations in appendix 11). At every gradient descent step (Fig.1.9.), we run a Lenia rollout with the current parameters (θ_l, A_l) and random obstacle placement (A_f) for 50 timesteps and apply a mean square error loss between the last state of the learnable channel (at last timestep) and a disk centered at the position of the goal we want to achieve. The gradient is then backpropagated through the Lenia steps to optimize both the parameters of the rule θ_l and the initialization A_l (details in appendix 11). As stated before, the obstacles are placed only on one side of a 256x256 grid. In total at every rollout 8 disk of radius 10 are randomly placed as obstacles.

Note that we filter from the history parameters leading to a collapse (mass reaches 0) and explosion of the pattern (pattern expanding too much) both when initializing the history with random parameters and also after an optimization loop (when the optimization fails) so that we do not use them as starting point for optimization in next steps. More details on the filter we applied can be found in appendix 11.

As presented before, our IMGEP outputs 160 parameters for each seed: 40 from the initialization of history and 120 from the IMGEP steps afterward (1 for each step). We discard the intermediate result of optimization and in each step of the IMGEP only save the final result of the optimization.

The initialization of the history plays an important role in the subsequent steps of the methods as all the following steps will be built on top of this basis, see Fig 3.a. We thus introduce an initialization selection in order to find promising initialization of the history. More details on this initialization selection mechanism can be found in appendix 7. Note that those steps are counted in the total number of lenia rollouts performed by the method for a fair comparison with random search, and are the main source of stochasticity in the number of rollout performed by a run of the method.

Robustness Evaluation. To measure the robustness of the agent against obstacles in the "basic obstacle test", we run 50 rollouts of 2000 timesteps with different obstacles positions. Each rollout environment has 23 obstacles of radius 10 randomly sampled uniformly in the whole grid and one placed in the trajectory of the moving agent (to be sure that it encounters obstacles), more details in appendix A.1.8. At the end of the 2000 timesteps, we compute statistics on the system rollout to detect if the matter is considered as an agent. We refer to appendix A.1.8 for more information on the statistics used for empirical agency and robustness tests. We then compute the ratio between the number of rollouts(ie environments) where the pattern survived (passed the empirical agency test) and the total number of rollouts. Robustness is measured similarly in the generalization tests but with 10 rollouts instead of 50. See appendix A.1.8 for more information on the different generalization tests.

Handmade search. The parameters from the original lenia papers [8, 9] are obtained from :

https://github.com/Chakazul/Lenia. We filter out the ones with multiple channels and the ones with an initialization that does not fit in the 256x256 grid, more details in appendix A.1.9.

1.2.5 Discussion

In closing this contribution, let us reiterate that what is interesting in such a system is that the computation of decision is done at the macro (group) level, showing how a group of simple identical entities can make "decision" and "sense" at the macro scale through local interactions only, and without a clear pre-existing notion of body/sensor/actuator. Seeing the discovered agents, it's even hard to believe that they are in fact made of tiny parts all behaving under the same rules. While some basic behavioural capabilities (spatially localized and moving entities) had already been found in Lenia with random search and basic evolutionary algorithms, this work makes a step forward showing how Lenia's low-level rules can self-organize robust sensorimotor agents with strong adaptivity and generalization to out-of-distribution perturbations.

Moreover, this work provides a more systematic method based on gradient descent, diversity search and curriculum-driven exploration to easily learn the update rule and initialization state, from scratch in high dimensional parameters space, leading to the systematic emergence of different robust agents with sensorimotor capabilities. We believe that the set of tools presented here can be useful in general to discover parameters that lead to complex self-organized behaviors.

Yet, several of the analyses we make in this work are empirical estimations or subjective. Future work shall consider how more formal definition(s) of agency and sensorimotor capabilities could be applied to the high-dimensional systems studied here[237, 245].

Also, engineering subparts of the environmental dynamics with functional constraints (through predefined channels and kernels) has been crucial in this work to shape the search process [266] towards the emergence of sensorimotor capabilities, as well as used as a tool to analyze more easily these emergent sensorimotor capabilities. An interesting direction for future work is to add even more constraints in the environment such as the need for food/energy to survive, the principle of mass conservation, or even the need to develop some kind of memory to anticipate future perturbations. We believe that richer environmental constraints and opportunities might be a great leap forward in the search for more advanced agent behaviors. For example, behaviors like competition between individuals/species for food, foraging or even basic forms of learning might emerge. From this competition and new constraints, interesting strategies could emerge as a form of autocurricula, as in [39, 266].

In fact, beyond individual capabilities, we could even wonder under what conditions one could observe the emergence of an open-ended evolutionary process [1] directly in the environment, without any outer algorithm, resulting in the emergence of agents with increasingly complex behaviors. To achieve this, we might need to use an optimization process similar to the one presented in this article to evolve all the

environmental rules instead of pre-specifying some of them by hand. Indeed, while the engineering of specific environmental rules facilitates the understanding/studying of the results, having more systematic ways to generate them could take us closer to the fundamental scientific quest of designing open-ended artificial systems with forms of functional life and agency "as it could be". Some preliminary studies are underway [267] as well as the next contribution of this chapter Sec.1.3.

Beyond those fundamental scientific questions, future work might also consider broader applications of this work for biology and AI. In biology, inferring low-level rules to control complex system-level behaviors is a key problem in regenerative medicine and synthetic bioengineering [268, 269]. In this regard, cellular automata offer an interesting framework to model, understand and control the emergence of growth, form and function in self-organizing systems. However, they remain abstract models: entities in the CA exist on a predefined grid topology whereas physical entities have continuous position and speed ; states in the CA are well-defined whereas it is not clear where and how information is processed in living organisms; rules in the CA operate at a predetermined scale whereas real-world processes operate at nested and interconnected scales. In AI, with the recent rise of web-deployed machine-learning models including large language models [270, 271], we are also faced with an increasing blurring of boundaries between the AI and the rest of the "environment" (human end-users and the web itself). It is hence central to understand how to measure emergent agency and cognition in those AI systems, as well as how to interact with them despite the extremely large input and behavioral spaces involved. In this regard we believe that environments like the one considered in this work can be useful to better inform the debate in much bigger models, as they are rich enough to support emergent agential behaviors while simple enough to study those questions explicitly. Far from trivial, transferring insights from the considered artificial systems to real biological systems or to very large AI systems is an exciting area of research with a potential broad range of medical and societal applications [272, 273].

In this contribution, we observed reproduction and multi-agent interactions within simulations. However, the agents were governed by identical rules, significantly limiting the potential for evolutionary processes to emerge, as agents were often mere replicas of each other. Moreover, as agents could grow indefinitely without consuming "resources", it is hard to introduce environmental constraints and pressures.

To overcome these limitations, the next section introduces Flow Lenia: a mass-conservative adaptation of Lenia. By enforcing mass conservation, it becomes easier to implement environmental constraints that influence agent behavior. More importantly, Flow Lenia enables the coexistence of multiple "species" of agents within the same grid, fostering competition for shared resources.

As a bonus, mass conservation simplifies the discovery of spatially localized patterns, which was a difficult task in this contribution requiring complex optimization of a carefully engineered objective.

1.3 Flow-Lenia: Towards open-ended evolution in cellular automata through mass conservation and parameter localization

Context

This contribution is the result from the continuation of the collaboration with Bert Chan (google deepmind Tokyo) in 2022-2023. It is also the result from the master internship of Erwan Plantec which I co-supervised.

The contribution has been presented at the Alife 2023 conference and got the **best paper award** :

- Plantec, E., Hamon, G., Etcheverry, M., Oudeyer, P. Y., Moulin-Frier, C., Chan, B. W. C. (2023). *Flow-Lenia: Towards open-ended evolution in cellular automata through mass conservation and parameter localization.* **In Artificial Life Conference Proceedings** 35 (Vol. 2023, No. 1, p. 131). MIT Press.
  Paper,Code

An extended version of conference paper has been accepted in the Alife Journal (invited contribution) and soon to be published.

We strongly encourage to take a look at the companion websites (link,link) to get a better view of the dynamic of the system with videos, as well as our contribution to the Virtual Creatures Competition 2024 (VCC2024): video.

Abstract

Central to the artificial life endeavour is the creation of artificial systems spontaneously generating properties found in the living world such as autopoiesis, self-replication, evolution and open-endedness. While numerous models and paradigms have been proposed, cellular automata (CA) have taken a very important place in the field, notably as they enable the study of phenomenons like self-reproduction and autopoiesis. Continuous CA like Lenia have been shown to produce life-like patterns reminiscent, from an aesthetic and ontological point of view, of biological organisms we call creatures. We propose in this contribution *Flow-Lenia*, a mass conservative extension of Lenia. We present experiments demonstrating its effectiveness in generating spatially-localized patterns (SLPs) with complex behaviors and show that the update rule parameters can be optimized to generate complex creatures showing behaviors of interest. Furthermore, we show that Flow-Lenia allows us to embed the parameters of the model, defining the properties of the emerging patterns, within its own dynamics, thus allowing for multispecies simulations. By using the evolutionary activity framework as well as other metrics, we shed light on the emergent evolutionary dynamics taking place in this system.

1.3.1 Introduction

An important challenge in artificial life (ALife) and artificial intelligence (AI) is about the design of systems displaying open-ended intrinsic evolution (i.e unbounded growth of complexity through intrinsic evolutionary processes) [274]. Such a process is called *intrinsic* since no final objective (i.e fixed fitness function) is set by the experimenter, the fitness landscape is intrinsic to the system and depends only on its current state, as in natural evolution where there is no final goal [31]. Seminal works by Von Neumann and Ulam in 1951 have paved the way in this direction. They were particularly interested in building an universal self-reproducing cellular automata (CA) capable of achieving open-ended evolution [275] quickly followed by Codd's attempt [276]. Further developments in this direction have quickly followed with Langton's self-replicating loops, a simpler model of self-replication in CA's, at the cost of its universality [70]. Even though, Langton's loops were able to self-replicate, no variations could be introduced in the process, thus making the emergence of evolution impossible. Fifteen years after Langton's self-replicating loops, the goal of obtaining an evolutionary process was achieved by Hiroki Sayama with the Evoloops model which displays Darwinian evolution of self-reproducing Langton's like loops [277] (see [278] for a more complete account of works on evolution and CA). Emergent evolutionary dynamics have also been studied in the context of neural cellular automata [88, 257] and artificial chemistry [72].

However, such systems rely on hand-defined rules, specific structures and controlled settings, ultimately limiting the diversity of patterns that can emerge in the system. On the other hand, even though Lenia creatures (Sec.1.1.2) display greater diversity, different creatures are governed by different update rules, and therefore cannot co-exist in the same world (i.e the same simulation) and cannot interact. Obtaining an evolutionary process in a CA could be achieved by embedding information in the system locally, modifying the update rule, and thus altering the properties of emerging creatures. This would act like a genome, enabling multi-species simulations. Such simulations might set the stage for evolution to occur in populations of patterns each with their own update rule and parameters. However, achieving it in a CA like Lenia is still an open problem.

One very important problem related to this objective is how can one actually measure these emergent evolutionary processes. Two main difficulties exist here. First, in such complex self-organized systems, fitness is intrinsic, i.e there is no externally nor well-defined fitness function, thus, one cannot have an objective measure of how adapted an individual is (if there is even a notion of individual). Evolutionary pressures are intrinsic to the system, where self-organised structures have to maintain their own integrity through cooperation or competition with other structures. Secondly, such a measure of evolution should be applicable to a wide range of systems and so rely on as few assumptions about the studied system as possible, fundamental desiderata if one's objective is to study life-as-it-could-be. The

framework of evolutionary activity [279, 280] proposed different measures aiming at discerning whether or not evolution is taking place, and quantifying it, in an observed system. Such measures have been applied to artificial systems such as Tierra [281] and Avida [282] and have even been used to compare their dynamics to real-world data [283]. Importantly, such a measure, or ensemble of measures, could allow to define a clear optimization objective for ALife researchers.

We believe that adding mass conservation is a key ingredient to address the aforementioned challenges. Such a constraint could (i) restrict emerging creatures to spatially localized ones, (ii) allow for the design of multi-species simulations and (iii) provide an important evolutionary pressure [284]. Conservation laws have been thought of as fundamental laws for Darwinian evolution to take place [285]. We propose in this work a mass-conservative extension to Lenia called *Flow-Lenia* and demonstrate that such conservation laws effectively facilitate the search for artificial creatures by constraining (almost all) emerging patterns to spatially localized ones. We also show that the update rule parameters can easily be optimized using vanilla evolutionary strategies [99] with respect to some fitness functions to obtain patterns with specific properties such as directed or angular motion. Importantly, we show that the Flow-Lenia formulation enables the integration of the parameters of the CA update rules within the CA dynamics, making them dynamic and localized, allowing for multi-species simulations, with locally coherent update rules that define properties of the emerging creatures. By describing trajectories of parameters over large timescales as well as by using measures of evolutionary activity [279, 280] and diversity, we evaluate the evolutionary dynamics emerging from these multi-species simulations. Moreover, we show that this system is relevant for testing ecological theories of evolution by studying two variations of the vanilla model, one based on dissipative dynamics, and one with resources that creatures need to consume for their survival. The study of the role of dissipative dynamics in this setting is motivated by dissipation having been proposed as one of the four pillars of "lyfe", a more general definition of life [286]. Introducing dissipative dynamics of the Flow-Lenia system could lead to the emergence of more interesting evolutionary dynamics characterized by higher evolutionary activity measures. On the other hand, resource limitations coupled to a shared pool of resources might create important selective pressures bootstrapping the intrinsic evolutionary process leading to higher evolutionary activity.

This contribution comes associated with a companion website `https://sites.google.com/view/flow-lenia` showing videos of the system dynamics, as well as open-source code directly executable in an online notebook [12]. 12:

1.3.2 Lenia

We refer to Sec.1.1.2 for a high-level description of the Lenia system whose dynamics are illustrated in Figure 1.14.a.

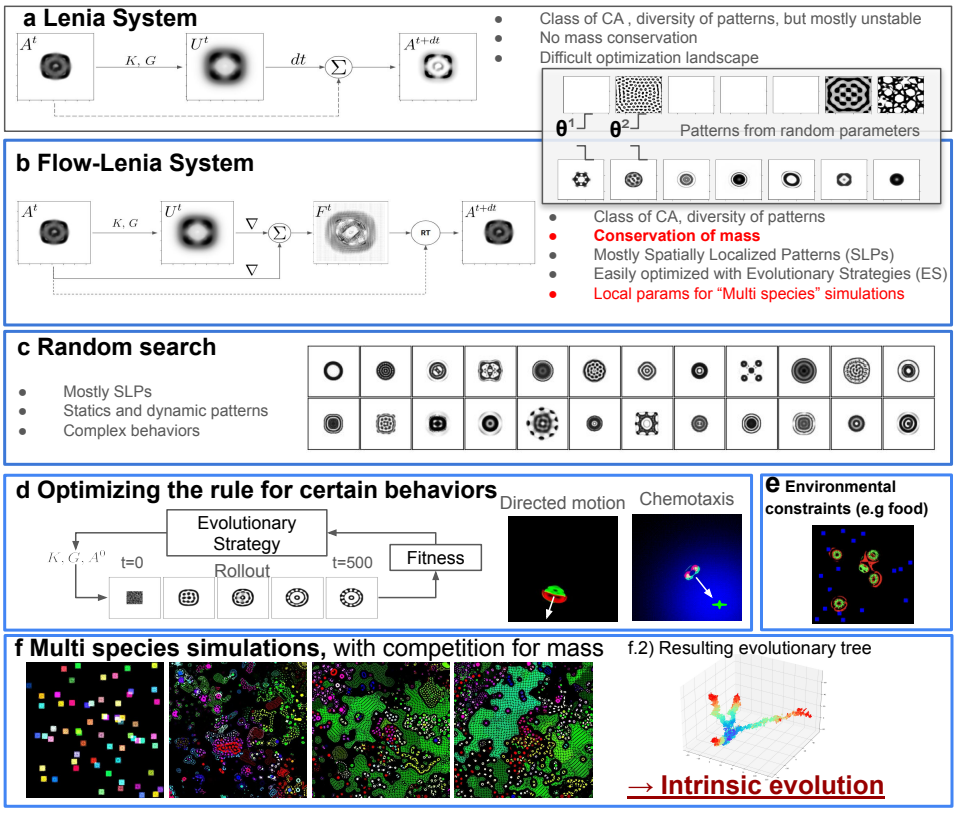


Figure 1.14: **Overview of the contribution.** We present *Flow-Lenia*, an extension of the Lenia (a) continuous Cellular Automata (CA). Flow-Lenia (b) introduces a built-in constraint for mass conservation, strongly facilitating the discovery of life-like patterns (c), the optimization of the system parameters towards certain behaviors (d) and the introduction of environmental constraints (e). Moreover, it allows to embed the system parameters within its own local dynamics, leading to large-scale multi-species simulations analysed in the light of the Evolutionary Activity framework (f). **(a) Lenia system**. The growth U^t is computed with kernels K and growth functions G. A small portion of the growth is then added to activations A^t to give the next state A^{t+dt}. (section 1.3.2). **(b) Flow-Lenia system**. U^t is computed as in Lenia and interpreted as an affinity map. The flow F^t is given by combining the affinity map and activation gradients. The next state is obtained by "moving" matter in the CA space according to the flow F^t using reintegration tracking. (section 1.3.3). Inset on the right of (a,b) shows 7 patterns obtained with randomly sampled update rules parameters, in Lenia (top, resulting mostly in non-SLP or empty patterns) and Flow-Lenia (bottom, resulting mostly in SLP patterns). **(c) Random search**. Patterns emerging from random parameter sampling in Flow-Lenia are qualitatively analyzed (sections 1.3.4 and 1.3.5). **(d) Optimizing the system update rule** is performed using simple evolutionary strategies with respect to predefined fitness functions, resulting in creatures with specific behaviors (e.g directed motion or chemotaxis). (sections 1.3.4 and 1.3.5). **(e) Environment constraints**. Example of environment with food (blue) that creatures can consume to gain mass. **(f) Multi species simulations**. Snapshots of a large scale multi-species simulation enabled by the parameter embedding mechanism (section 13), resulting in a evolutionary tree (f.2) (section 1.3.5).

We here shortly describe the Lenia update and its notations. For a more detailed explanation, see [8, 9]. Let $\mathscr{L}$ be the support of the CA, here a two-dimensional grid defining the set of cells as well as their spatial relationships. The state of the Lenia system at time t is then defined by the map $A^t : \mathscr{L} \to [0,1]^C$ where C is the number of channels of the system. The system update rule is then defined by the tuple $< K, G, c_1, c_0, A^0 >$ where K is a set of convolution kernels with $K_i : \mathscr{L} \to [0,1]$ satisfying $\int_{\mathscr{L}} K_i = 1$ and G is a set of growth functions with $G_i : [0,1] \to [-1,1]$. Each pair (K_i, G_i) is associated with a source channel c_0^i it senses and a target channel c_1^i it updates. Connectivity can be represented through a square adjacency matrix M of size C where $M_{ij} \in \mathbb{N}$ is the number of kernels sensing channel i and updating channel j. A^0 is the initial state of the system. We use the same kernels as the one used in the previous contribution Sec.1.2 and detailed in Appendix.A.1.6 . In this version, kernels are radially

symmetrical and defined as a sum of concentric Gaussian bumps :

$$K_i(x) = \sum_{j=1}^{k} b_{i,j}\, exp\left(-\frac{(\frac{x}{r_i R} - a_{i,j})^2}{2w_{i,j}^2}\right) \tag{1.1}$$

Where a_i, b_i, w_i and r_i are parameters defining kernel i. k is a parameter defining the number of rings per kernel (set to 3 here) and R is a parameter common to all kernels defining the maximum neighborhood radius. Each kernel is then defined by $3{\times}k{+}1$ parameters. Growth functions are defined as Gaussian functions scaled in the range $[-1, 1]$:

$$G_i(x) = 2\, exp\left(-\frac{(\mu_i - x)^2}{2\sigma_i^2}\right) - 1 \tag{1.2}$$

Where μ_i and σ_i are parameters of growth function i so each growth function is defined by 2 parameters. A step in Lenia is defined by the following steps (see figure 1.14 (top)) :

1. Compute the growth at time t given the actual state A^t :

$$U_j^t = \sum_{i=1}^{|K|} h_i \cdot G_i(K_i * A^t_{c_0^i}) \cdot [c_1^i = j] \tag{1.3}$$

   Where $h \in \mathbb{R}^{|K|}$ is a vector weighting the importance of each pair (K_i, G_i) and $[c_1^i = j]$ is the Iverson bracket which equals 1 if $c_1^i = j$ and 0 otherwise (i.e equals 1 if the ith pair updates channel j).
2. Add a small portion of the growth U^t to the actual state A^t to get the state at the next time step and clip results back to the unit range :

$$A_i^{t+dt} = [A_i^t + dt\, U_i^t]_0^1 \tag{1.4}$$

1.3.3 Flow-Lenia

Flow-Lenia extends the Lenia system in the sense that it reuses all the aforementioned components. We propose for this system to interpret activations as concentrations of "matter" in all cells and to refer to the term U^t, previously called the growth in Lenia, as an affinity map. The idea is that the matter will greedily move towards higher affinity regions by following the local gradient of the affinity map U, $\nabla U : \mathscr{L} \to \mathbb{R}^2$. To do so, we define a flow $F : \mathscr{L} \to (\mathbb{R}^2)^C$, which can be interpreted as the instantaneous speed of matter, as:

$$\begin{cases} F_i^t = (1 - \alpha^t)\nabla U_i^t - \alpha^t \nabla A_\Sigma^t \\ \alpha^t(x) = [(A_\Sigma^t(x)/\beta_A)^n]_0^1 \end{cases} \tag{1.5}$$

With $A_\Sigma^t(x) = \sum_{i=1}^{C} A_i^t(x)$ the total mass in each location x. Here ∇U_i^t is the affinity gradient for channel i. The negative concentration gradient $-\nabla A_\Sigma^t$ is a diffusion term to avoid concentrating all the matter in very small regions akin to the clipping in Lenia which upper bounds concentrations. In practice, gradients are estimated through Sobel filtering. The map $\alpha : \mathscr{L} \to [0, 1]$ is used to weight the importance of each term such that $-\nabla A_\Sigma^t$ dominates when the total mass at a

given location is close to a critical mass $\beta_A \in \mathbb{R}_{>0}$. Intuitively, the result is that matter is mainly driven by concentration gradients in high concentration regions and is more free to move along the affinity gradient in less concentrated areas. We typically use $n > 1$ such that the affinity gradient dominates on a larger range of masses.

Finally, matter can be displaced in space according to flow F giving us the state at the next time step. To do so we use the reintegration tracking method proposed in [287]. Reintegration tracking is a semi-Lagrangian grid based algorithm thought as a reformulation of particle tracking in screen space (i.e grid space) aimed at not losing information (i.e particles) which happens when two particles end up in the same cell. The basic principle is to work with a distribution of particles (i.e infinite number of particles) and conserve the total mass by adding up masses going to the same cell. Overall, reintegration tracking can be seen as a grid-based approximation to particle systems with an infinite number of particles, having the property to conserve total mass. Thus, Flow-Lenia can be seen as a new kind of model at the frontier between continuous CA and particle systems. A particle based model directly inspired by the Flow-Lenia formulation has been recently proposed in Mordvintsev, Niklasson, and Randazzo. Figure 1.15 illustrates how reintegration tracking is used in our case. The resulting update rule (animated in this video [13]) is the following :

13: https://sites.google.com/view/flowlenia/model

$$\begin{cases} A_i^{t+dt}(x) = \sum_{x' \in \mathscr{L}} A_i^t(x') I_i(x', x) \\ I_i(x', x) = \int_{\Omega(x)} \mathscr{D}(x_i'', s) \end{cases} \tag{1.6}$$

With $x_i'' = x' + dt \cdot F_i^t(x')$ the target location of the flow from x' in channel i. $\Omega(x)$ is the domain of the cell at location p, which is a square of side 1. $\mathscr{D}(m, s)$ is a distribution defined on $\mathscr{L}$ with mean m and variance s satisfying $\int_{\mathscr{L}} \mathscr{D}(m, s) = 1$, which is in practice a uniform square distribution with side length $2s$ centered at m. This distribution emulates a flow of particles from the source area $\Omega(x')$ to the target area $\mathscr{D}(x'', s)$, where the distribution $\mathscr{D}$ emulates Brownian motion at the low level. s is a hyperparameter of the system which can be seen as a form of temperature. The reintegration tracking method is depicted in Fig. 1.15. Since the distribution $\mathscr{D}$ integrates to 1, it is clear that a cell cannot send out more mass than it contains nor less and so the system conserves its total mass. For computational reasons, we do not look at all cells to compute incoming matter as described by equation 1.6 but only at the neighborhood composed of cells whose Chebyshev distance to the target cell is less than 5 (extended Moore neighborhood) allowing for considerably reduced computation times. Mass conservation also implies that cells' states are no longer bound to the unit range but can be any positive real-valued number ($S \equiv \mathbb{R}_{\geq 0}^C$). This model has been implemented in JAX [289] allowing fast simulation on GPU ($255\mu s \pm 3.11\mu s$ per step on Tesla T4 GPU with 1 channel, 10 kernels, and 128×128 world size).

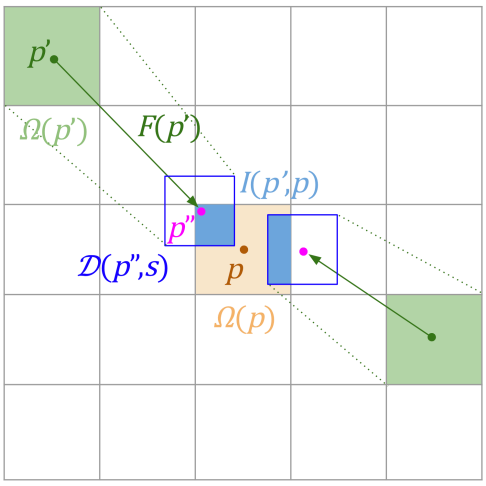


Figure 1.15: Calculation of incoming matter to cell $p \in \mathscr{L}$ through reintegration tracking [287]. Mass contained in cell at location $p' \in \mathscr{L}$ is moved to a square distribution $\mathscr{D}$ centered on $p'' = p' + dt \cdot F^t(p')$. The proportion of mass from p' arriving in p is then given by the integral of $\mathscr{D}$ on the cell domain of p, $\Omega(p)$, denoted as $I(p', p)$.

Flow-Lenia with parameters embedding

Flow-Lenia formulation, by considering a flow of matter, allows to attach any information to the moving matter such as the update rule parameters making them dynamic and localized. Formally, this comes to define a parameter map $P : \mathscr{L} \to \Theta$ where Θ is the parameter space. In this work, only the kernel weighting vectors h are included in the parameter space $\Theta \equiv \mathbb{R}^{|K|}$. This map can then be used to compute the affinity score in each cell x by weighting the influence of each pair (K_i, G_i) with the localized vector $P(x)$ giving the following formula:

$$U_j^t(x) = \sum_{i=1}^{|K|} P_i^t(x) \cdot G_i(K_i * A_{c_0^i}^t)(x) \cdot [c_1^i = j] \tag{1.7}$$

While in theory, all the parameters could be embedded in the parameter map, this would come with high memory and computational costs for some. In particular, changing the kernels parameters dynamically would make the use of fast convolution operations such as fast-Fourier convolution impossible as it would require using different kernels in all different locations of the map.
We can now move the parameters along with the matter during the reintegration tracking phase. This necessitates deciding what to do when different sets of parameters arrive in a same cell. We propose two different methods which are respectively *average* and *softmax sampling*. The former makes a weighted average of incoming parameters with respect to the quantities of incoming matter and is formally defined as :

$$P^{t+dt}(x) = \frac{\sum_{x' \in \mathscr{L}} A^t(x') I(x', x) P^t(x')}{\sum_{x' \in \mathscr{L}} A^t(x') I(x', x)} \tag{1.8}$$

Softmax sampling, on the other hand, samples a parameter in the set of incoming ones following the softmax distribution given by incoming quantities of matter :

$$\mathbb{P}[P^{t+dt}(x) = P^t(x')] = \frac{e^{A^t(x') I(x', x)}}{\sum_{x'' \in \mathscr{L}} e^{A^t(x'') I(x'', x)}} \tag{1.9}$$

Intuitively, the more represented set of parameters has a greater probability of being selected in the cell, like simulating in one step a competition between different parameters in the cell.

In the rest of this contribution, we use the softmax mixing rule. We chose this rule because of the competitive dynamics it creates in the system. It enables creatures to convert mass from other creatures with different parameters (i.e other species). This would not be possible with the average rule as each interaction would create a new set of parameters. In addition, the average rule tends to uniformize the parameters in the simulation.

1.3.4 Experimental methods

The experiments are divided into three main parts. First, we perform random search in the Flow-Lenia parameter space allowing us to qualitatively analyze the dynamics of the system and the typical patterns emerging from it. In a second part, we optimize the update rules parameters as well as the initial pattern configuration in order to obtain creatures displaying specific behaviors. Finally, we experiment with the parameters embedding mechanism and analyze the long-term temporal dynamics emerging from these multi-species simulations. Each of these experiments is explained in detail in sections 1.3.4, 1.3.4 and 1.3.4 respectively, and the associated results are presented in section 1.3.5.

Random search experiments

We performed random search in the Flow-Lenia parameter space described in table 1.1. We refer the reader to sections 1.3.2 and 1.3.3 for further details on the role of these parameters. Associated results are presented in section 1.3.5.

Initial patterns A^0 are set with a 40×40 patch with matter drawn from a uniform distribution in the center of the grid and no matter everywhere else.

Neighborhood			Growth functions		
R	$\in [2, 25]$		μ	$\in [0.05, 0.5]$	*
r	$\in [0.2, 1]$	*	σ	$\in [0.001, 0.2]$	*
Kernels			**Flow**		
h	$\in [0, 1]$	*	s	0.65	
a	$\in [0, 1]^3$	*	n	2	
b	$\in [0, 1]^3$	*	dt	0.2	
w	$\in [0.01, 0.5]^3$	*			

Table 1.1: Flow Lenia explored parameter space. Parameters marked with a * must be sampled for each kernel-growth function pair.

Directed search experiments

We used evolutionary strategies [99] to optimize the update rule parameters and the initial configuration A^0. We trained the model with respect to four different user-defined fitness functions, i.e tasks: directed motion, angular motion, navigation through obstacles and chemotaxis. We refer the reader to the appendix.A.2.1 for further details about the employed fitness functions.

We used EvoSax [290] implementation of the OpenES [99] strategy with a population size of 16 and Adam optimizer [291] with 0.01 as learning rate. We optimized the Flow Lenia update rule with different numbers of kernels and either 1 or 2 channels. For comparison, we also trained the original Lenia on the directed motion task following the same optimization procedure. The initial pattern is composed, as in random search, of a square patch with non-zero activations placed at the center of the world and zeros everywhere else.

Intrinsic evolution experiments

In order to analyze the potentially evolutionary dynamics enabled by the parameters embedding mechanism presented in section 13, we performed simulations with larger spatial and temporal scales. A similar attempt using the original Lenia system for large-scale simulation of intrinsic evolution was described in [267]. By allowing multispecies simulations, the parameters embedding mechanism also allows for interspecies competition especially under the stochastic parameter selection rule described in equations 1.7 as it allows species to convert matter from other species. We further propose two variations of the vanilla model, namely the dissipative and food models. During simulation, the set of unique parameters denoted $\mathscr{P}^t \equiv \{P^t(x)\}_{x\in\mathscr{L}}$ is recorded. Together with parameters, we record their associated total mass through time $M(p,t) = \sum_{x\in\mathscr{L}} A^t_\Sigma(x) \cdot [P^t(x) = p]$. We also introduce a diversity metric $D(t)$ quantifying the diversity of parameters and which is defined as the average distance between all present parameters in the system at this time t:

$$D(t) = \frac{1}{|\mathscr{P}^t|} \sum_{p\in\mathscr{P}^t} \sum_{p'\in\mathscr{P}^t} \|p - p'\|_2 \tag{1.10}$$

Where $\|\cdot\|_2$ is the euclidean norm. We only recorded simulation data every 100 steps for memory reasons. However, as interesting creatures' behaviors unroll in around 100 time steps, evolutionary dynamics must happen on much larger scales. The simulation settings as well as the three different models are described in more detail in the following sections.

Simulation settings. All presented models have been simulated for $500 \cdot 10^3$ steps. The system is initialized with, when not stated otherwise, 3 channels and 5 kernels per channel pair making a total of 45 kernels. We introduce mutations in the form of square "beams" affecting a random 10×10 patch in the grid. Beams apply a perturbation sampled from a normal distribution with mean 0 and unit variance to the parameter map P, the perturbation being the same for all cells in the affected patch. While we could have implemented mutations on single cells only, it would have been unlikely for any mutation to have the opportunity to develop as they would be quickly overtaken by their neighbors. Affecting larger zones using beams gives a mutation better chances of developing. Mutation rates are controlled by the parameter p_{mut} which is the probability of a mutation beam appearing at each time step. All simulations have been repeated with 5 different random seeds.

Model variations. We propose in this work three different variations of the Flow-Lenia model, namely: vanilla, dissipative and food which are presented hereafter.

Vanilla. In this setting, the environment is simply initialized with 64 creatures. Each creature is initialized as a 20×20 square patch which position is uniformly sampled on the grid $\mathscr{L}$. Matter concentrations (A) in these patches are also sampled uniformly in $[0, 1]$ and parameter (P) is sampled following a normal distribution and set identically for all cells in a patch.
In this setting, high fitness parameters are the ones leading to creatures able to preserve and increase their associated mass. Note that here, creatures can only grow by converting matter from other creatures. This creates pressures for strong individuality, especially with the stochastic update rule, as it incentivizes creatures to protect their resources. But strategies that are too defensive, preventing a parameter set from expanding (i.e gaining mass and territory), might put it at risk as it might make it more vulnerable to disappear because of a random mutation beam.

Dissipative. In the dissipative setting, the world is initialized in the same way as the vanilla setting and mutations are also used in the same way. The difference is that in this setting we regularly remove matter and the associated parameters and add new ones, thus creating dissipative dynamics. To do so, we use two new types of beams, one removes matter and parameters in the affected patch, the other adds a new creature (i.e a patch with random parameters, as in the initial pattern) at another affected location sampled in a 100×100 corner of the grid, the input zone, with randomly initialized parameters. The new creature is initialized in the same way as the initialization phase. The rate at which the dissipative beams appear is controlled by parameter p_{diss}.
We expect the dissipative setting to create more interesting evolutionary dynamics characterized by higher evolutionary activity measures (see section 1.3.4) by creating an environment with a constant input of novelty in the form of new parameters while conserving more stable zones in the environments (i.e the ones further from the input zone).

Food. In this last setting, we introduce an additional "food" mechanism where creatures would need to collect resources in order to replenish their own constantly decaying pool of resources. To do so, we let matter decay at a fixed rate ρ_{decay}, and create a "food" map $\Psi : \mathscr{L} \to [0, \infty)$. When matter is in a cell where there is also food, then food is transformed into matter at a given rate ρ_{digest} giving the following update.

$$\begin{cases} A^{t+dt}(x) = \cdots + [A^t(x)\rho_{digest}]_0^{\Psi^t(x)} - A^t(x)\rho_{decay} \\ \Psi^{t+dt}(x) = \Psi^t(x) - [A^t(x)\rho_{digest}]_0^{\Psi^t(x)} \end{cases} \tag{1.11}$$

Where $\cdots$ refers to the update equation 1.6 and $[\cdot]_a^b$ is the clip function between a and b. We enable creatures to sense food by adding kernels and growth function from the food map Ψ to creatures' channels A.

The food map is initialized with 32 5×5 food squares (where the value of the map is set to 1 for all cells in the patch) randomly sampled on

the grid. At each time step, a new food patch is added with probability p_{food}.
In this model, a high fitness creature is one able to counter its constant decay by either converting matter from other creatures (i.e being a predator), or consuming food resources. Such a constraint for creatures, creating a need to find food for their continued existence, together with a common pool of resources, might create strong evolutionary pressures and competitive dynamics, bootstrapping the emergence of evolutionary processes in the system. Moreover, the addition of the food constraint and the necessity to counter-act their decay introduce the notion of a minimal criterion, i.e a criterion creatures must met in order to expand, which has been proposed as a fundamental ingredient for open-ended evolution to emerge [5, 292].

Measuring evolutionary activity. Multiple measures of evolutionary activity have been proposed in the literature. In this work, we use two different measures, namely count-based evolutionary activity (EA^C) and non-neutral evolutionary activity (EA^N) [280]. Evolutionary activity metrics are based on records of the presence and counts of different components in a system. Components can be for instance molecules or species. In our case, the components are the different parameters, meaning that a component, or species, is a unique point in the parameter space. Hence, two sets of parameters with infinitely small differences are considered as different species, regardless of their phenotypic outcome. This is a current limitation of the study as discussed in the section 1.3.6. We here use both parameter sets $\mathscr{P}^t$ and associated masses M to compute component level activities $a_p^C(t)$ and $a_p^N(t)$ for count-based and non-neutral activities respectively. Count-based activity is based on the total mass associated with a given set of parameters. At each time step, the count-based activity of a component is incremented by its total associated mass if this component exists:

$$a_p^C(t) = (a_p^C(t-1) + M(p,t)) \cdot [p \in \mathscr{P}^t] \tag{1.12}$$

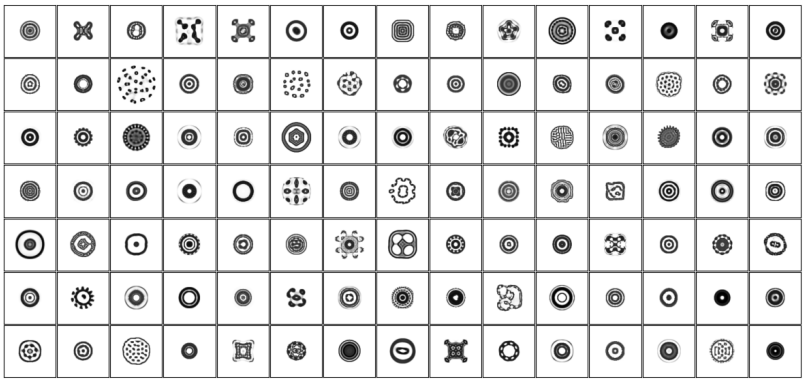
Where $[x]$ is the Iverson bracket which equals 1 if x is true and $M(p,t)$ is the mass associated to species p at time t.
Intuitively, the greater the total mass of a parameter p is, and the longer it survives, the higher will be its activity a_p. While count-based activity can give useful insights, it does not tell much about the quantity of change happening in the environment. Non-neutral activity solves this issue by penalizing periods of stasis. Here, the activity of a component is incremented by the square of the change of its proportion in the population of components if it increased. Hence, if multiple components stay very stable, i.e reach a stable equilibrium, their activities will remain constant. However, if a component sees its proportion in the population going up then its activity will increase. This measure thus prevents periods of stasis from contributing to the evolutionary activity measure. This is formally defined by the following set of equations:

$$\begin{cases} a_p^N(t) = (a_p^N(t-1) + \Delta_p^N(p,t)) \cdot [p \in \mathscr{P}^t] \\ \Delta_p^N(t) = (\sum_{p'} M(p',t)) \cdot (\rho(p,t) - \rho(p,t-1))^2 \cdot [\rho(p,t) > \rho(p,t-1)] \\ \rho(p,t) = \frac{M(p,t)}{\sum_{p'} M(p',t)} \end{cases} \tag{1.13}$$

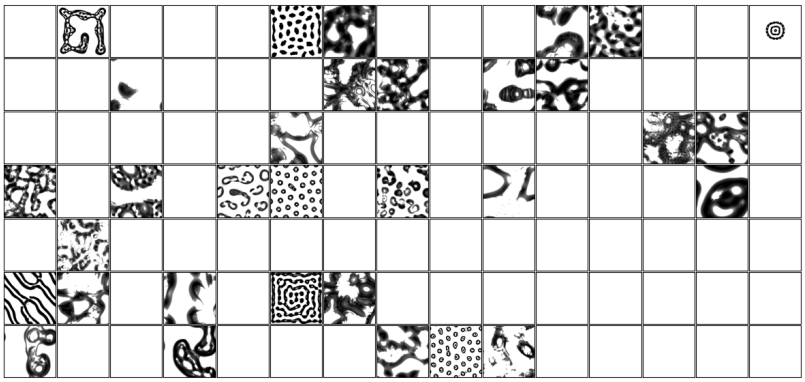
Global activity at a given time is simply defined as the sum of all components activities at this same time step: $EA^*(t) = \sum_p (a_p^*(t))$ where $*$ is either C or N for count-based and non-neutral evolutionary activities respectively.

1.3.5 Results

Random search



(a) Flow-Lenia



(b) Lenia

Figure 1.16: Patterns obtained from 105 different randomly sampled update rule parameters in (a) FLow-Lenia and (b) Lenia systems. Each pattern is obtained by simulating the systems for 150 steps from an initial state composed of a 40×40 patch with uniformly sampled concentrations. The exact same 105 parameter sets are used for both systems.

By performing random and manual search of the Flow Lenia parameter and hyperparameter space described in table 1.1 we have been able to discover SLPs with already interesting and complex behaviors some of which are displayed in figures 1.16 and 1.17.
Most of the patterns generated in Flow Lenia are SLPs (see figure 1.16(a)) with rare exceptions found by manually setting parameters to specific configurations leading to scattered matter. We can see in figure 1.17(b) that the same parameters mostly lead to empty or exploding patterns in the Lenia system. Using multiple kernels led to the emergence of SLPs with more complex shapes and behaviors. While

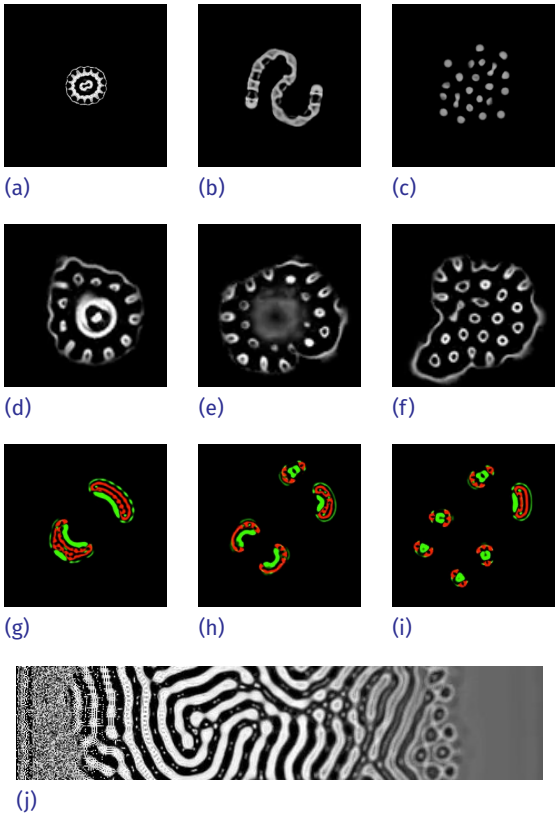


Figure 1.17: Flow Lenia creatures. (a-c) Samples of creatures found through random search in Flow Lenia parameter space. (d-f) and (g-i) Timelapses of patterns found through random search. Colors in (g-i) code for different channels. (j) Effect of changing temperature in Flow Lenia, temperature is linearly increasing from left to right. Videos are available at *https://sites.google.com/view/flow-lenia*.

part of emerging patterns tend to be static ones, dynamic patterns are quite common in Flow Lenia. For instance gyrating SLPs (Fig. 1.17.a) or snake like patterns (b) with complex motion emerging from attraction/repulsion dynamics can be frequently observed. Dividing and merging dots (c) resembling reaction-diffusion patterns are also a common pattern. Timelapse (d-f) shows a creature with complex and unpredictable dynamics emerging from the interactions of its membrane, multiple organoids-like structures and a central nuclei ultimately leading to a phase transition happening in (e). Timelapse (g-i) shows a 2-channels creature displaying complex division patterns and interesting modular creatures whose characteristics change depending on their total mass while being of the same "kind" (i) (see 5 creatures on the leftmost part of (i)). Note that multi-channel creatures often show more complex dynamics and patterns with very modular shapes where each channel seems to occupy a different role. (j) shows the effect of changing the size of the reintegration tracking distribution s (see equation 1.6 and figure 1.15), a parameter we call temperature. Here temperature is linearly increasing from left to right showing very different phases of the systems. More interestingly, patterns at the frontier between the Turing-like phase (center) and the equilibrium phase (right) are much more dynamic and display unpredictable dynamics suggesting a critical regime.

Small scale simulation with food. By performing short simulations with random sets of parameters, parameter embedding, and food, we have been able to observe interesting patterns. First, we have been able to observe that some creatures, while not having trained for it,

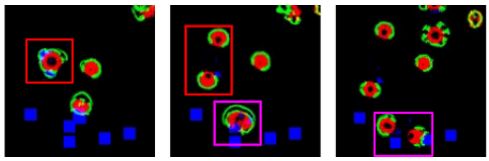


Figure 1.18: Timelapse of simulation with parameter embedding and food (in blue) showing division events (highlighted with boxes). Videos are available at this link.

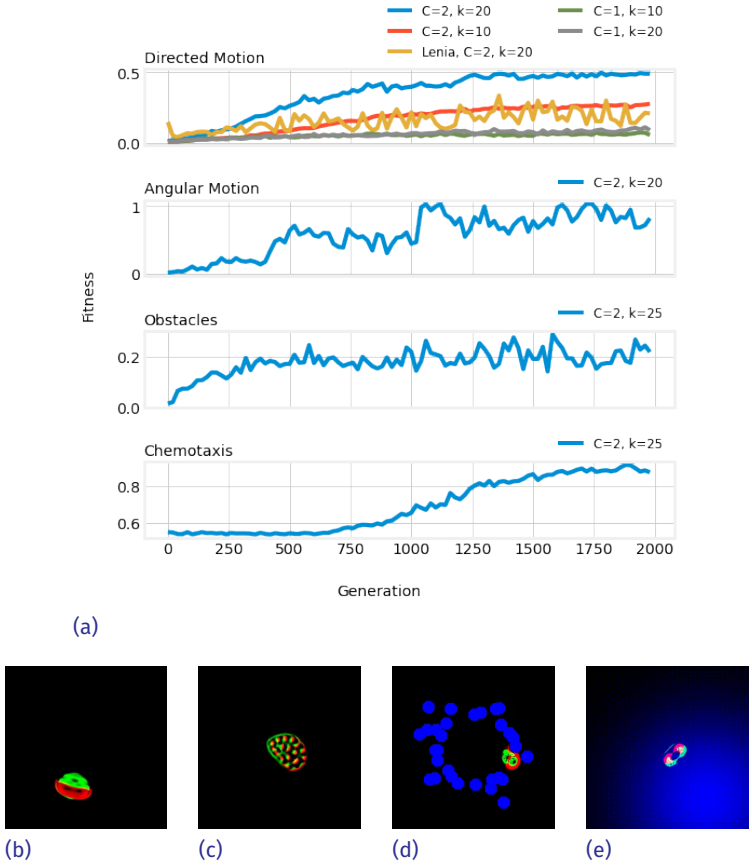


Figure 1.19: (a) Results of evolutionary optimization. C is the number of channels of the system and k is the number of kernels and growth functions. When performing the exact same optimization for directed motion in the original Lenia system (yellow curve), not only optimization is unstable but it only discovers exploding patterns. (b-e) Creatures found through optimization. (b) Directed motion with 2 channels and 20 kernels. (c) Angular motion with 2 channels and 20 kernels. (d) Motion through obstacles with 2 channels and 25 kernels. (e) Chemotaxis with 2 channels and 25 kernels. Videos are availabe at `https://sites.google.com/view/flow-lenia`.

are able to go towards nearby food sources and consume them. We can hypothesize that creatures with such a capability will survive (and grow) while others will not, leading to intrinsic evolution. Quite interestingly, complex patterns can emerge from the change of mass induced by decay or food consumption. For instance, when growing after eating, some creatures will divide into two identical creatures as shown in figure 1.18 (f), a crucial pattern for evolution to occur. On the other hand, mass decay also leads to interesting dynamics where creatures undergo phase transitions, changing their shape and behavior, when their mass falls below a certain threshold which can lead them to adopt foraging behaviour for example while being initially static.

Optimizing Flow Lenia creatures

Flow-Lenia update rule parameters can also be easily optimized so to generate patterns with specific behaviors. This is a difficult task in Lenia as it would require constantly monitoring the existential status and the spatially-localizedness of evolved creatures. Thus, training creatures in Lenia requires to define characterizations of creatures accounting for such properties which is a far from trivial problem. Moreover, even if one can come up with proxies to find spatially localized patterns, the optimization process remains difficult necessitating advanced optimization methods like curriculum learning used in the previous contribution Sec.1.2. In Flow Lenia, the spatial localization constraint is intrinsic to the system thus removing the necessity to

account for it when searching for creatures.
Using evolutionary strategies [99] we have been able to find creatures solving various tasks such as:

Directed motion. The creature is able to move as fast as possible in one direction. Efficient solutions can be found in the 2 channels condition but not in the single channel case. However, when running the optimization algorithm for longer (e.g 5000 generations), we have been able to find single channel creatures with similar fitness than their 2 channels counterpart. Increasing the number of kernels led to faster discovery of good solutions. The best performing creature is shown in figure 1.19(b). This creature moves because of attraction/repulsion dynamics between the 2 channels which might explain why directed motion is much easier to attain with multi-channels creatures. On the other hand, the optimization of the original Lenia model is much less stable and discovered patterns are less successful than their mass-conservative counterparts. Moreover, every Lenia optimized patterns are exploding ones.

Angular motion. The creature is able to maximize its straight line speed as well as to make turns. The best performing creature, shown in figure 1.19 (c), displays very complex internal dynamics leading it to periodically make 180° turns while moving in a straight line the rest of the time. These dynamics seem to be generated by attraction-repulsion dynamics like the ones observed in directed motion but here in a more intricate morphology.

Navigation through obstacles. The creature is able to maximize its traveled distance while multiple obstacles are placed on its way. We have been able to successfully train creatures able to move and maintain their integrity when making contact with walls such as the one shown in figure 1.19 (d) which is able to resist deformation and find a way out of the "forest". In comparison, solving a similar task in Lenia required complex optimization methods based on curriculum learning, diversity search and gradient descent over a differentiable CA in Sec.1.2. However, such a comparison is difficult because Flow Lenia creatures are inherently more robust due to the conservation of mass, whereas Lenia creatures can disappear because of perturbations.

Chemotaxis. The creature is able to follow a concentration gradient, encoded in a separate channel and which it is able to sense through some kernels, towards its source. The best solutions such as the one shown in figure 1.19 (e) are perfectly able to climb the gradient towards its maximum.

For further details on the optimization procedure, we refer the reader to the appendix.A.2.1.

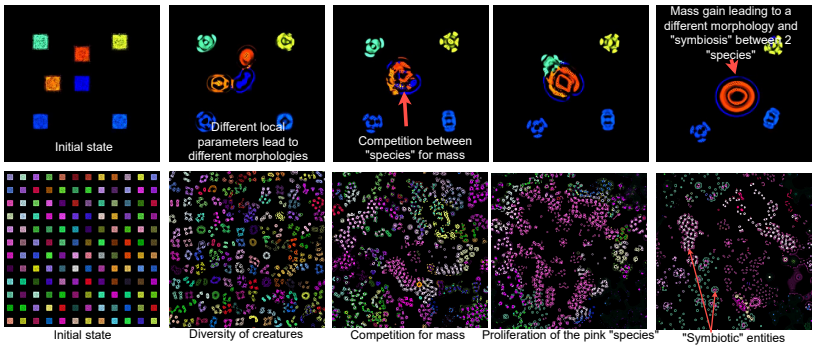


Figure 1.20: Emergent evolution in a multi species simulation in Flow Lenia. Video at *https://sites.google.com/view/flowlenia/*

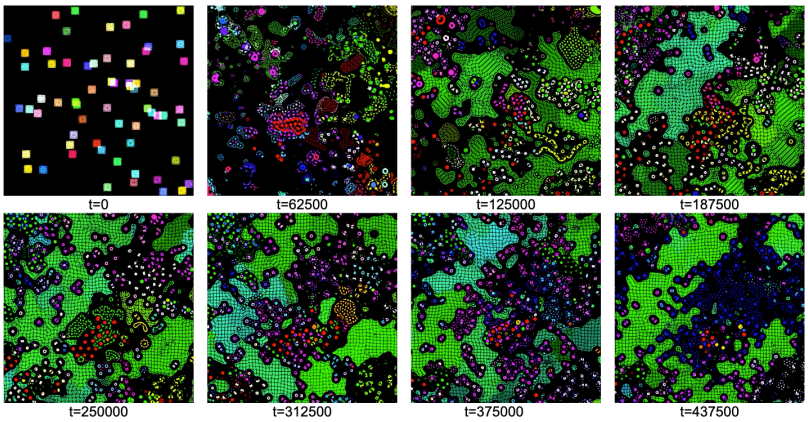


Figure 1.21: **Vanilla** Snapshots of simulations for the vanilla model. Colors are defined by the parameter map while intensity is set by concentrations of matter. The snapshots shows a very large and stable green structure, instance of a larger scale creature. Videos of different simulations are available in the associated website *https://sites.google.com/view/flow-lenia*

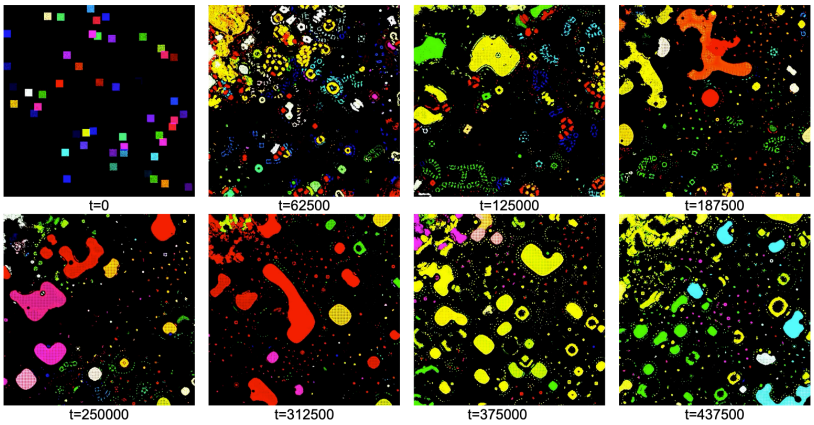


Figure 1.22: **Dissipative** Snapshots of simulations for the , dissipative model. Colors are defined by the parameter map while intensity is set by concentrations of matter. Videos of different simulations are available in the associated website *https://sites.google.com/view/flow-lenia*

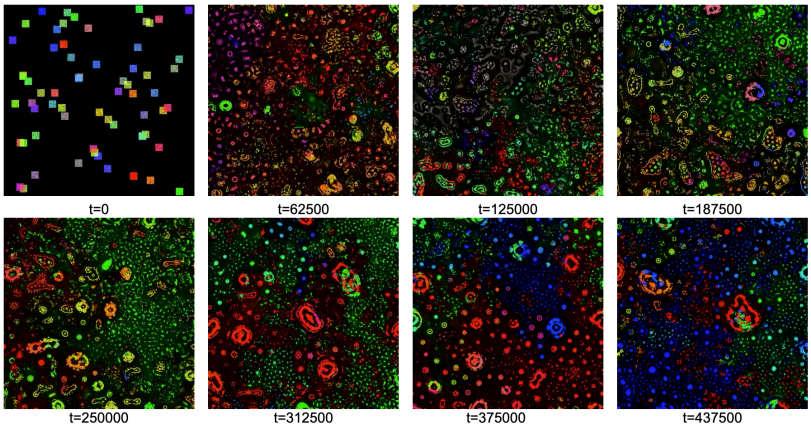


Figure 1.23: **Food** Snapshots of simulations for the food model. Colors are defined by the parameter map while intensity is set by concentrations of matter. Videos of different simulations are available in the associated website *https://sites.google.com/view/flow-lenia*

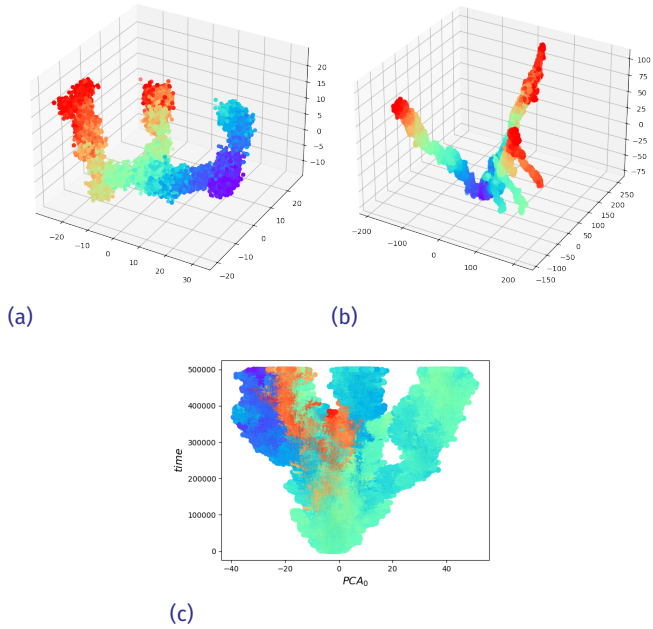


Figure 1.24: Visualization of Flow-Lenia evolutionary trajectories through projection in the parameter subspace formed by the principal components of the set $\mathscr{P}$ of parameters having existed during the simulation. (a) and (b) show two different evolutionary trees obtained from simulations of the food model (a) and vanilla model with deterministic parameter mixing rule (b). Colors are coding for time (from purple to red) showing for instance two branches having survived and one which went extinct in (a). (c) is an alternative visualization of the data in 2 dimensions where the x axis is the first principal component, the y axis is the time axis, colors code for the second principal component obtained from a simulation with the vanilla model.

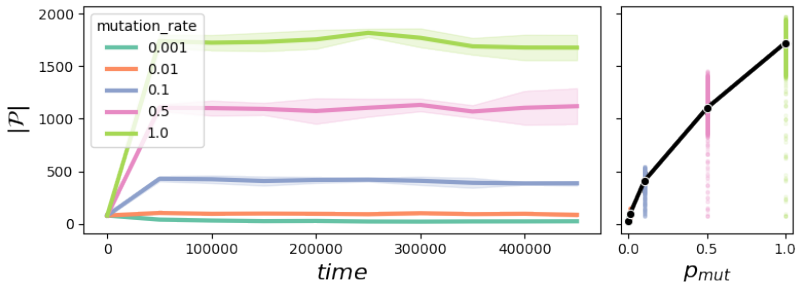


Figure 1.25: Evolution of the number of different parameters through time for different mutation rates p_{mut}. (left) The number of different parameters is plotted against time where error bands correspond to the standard deviation over 5 different runs. (right) The average number of different parameters over time is plotted against p_{mut}.

Intrinsic evolutionary dynamics

Snapshots of sampled simulations can be seen in figure 1.20 1.21,1.22,1.23. In all settings we tried, we visually observed changes in the set of parameters present in the environment through time, often with some species taking over others, leading to extinctions, and mutations giving rise to new species able to survive. We also observe interesting interspecies dynamics where different parameters form stable structures without competing which can be seen as some form of cooperation or symbiosis. It should be noted that the general appearance of simulations highly depends on the parameters of the system, i.e the kernels and growth function parameters which stay fixed during simulation. While some simulations are visually appealing to us, displaying creatures with different scales and behaviors, other might look much more chaotic with only very small dot-like creatures creating poorly human-readable dynamics.

In order to analyse how the parameters evolve through time and move in the parameter space, we first obtained the subspace of maximal variation of the parameter space through Principal Component Analysis (PCA). Principal Components were fitted with the complete set of parameters having existed through simulation $\mathscr{P} \equiv \cup_{0 \leq t \leq T} \mathscr{P}^t$, where $\mathscr{P}^t$ is the set of parameters present in the world at time t. Then,

we can visualize trajectories in the parameter space by projecting the set of parameters present at each timestep $\mathscr{P}^t$ in this subspace. Example of trajectories are shown in figure 1.24. Interestingly, we can notice that motion in this space looks far from random and take the form of a tree, an evolutionary tree. We can see the formation of different branches which could be seen as instances of speciation. This differentiation clearly indicates the presence of an intrinsic fitness landscape. We also observe that trees obtained from simulations with the deterministic sampling rule for parameters (see sections 13) have much thinner branches and distinctive trajectories. Interestingly, the stochastic sampling rule only adds more stochasticity in the interactions between species, not on the parameters, indicating that these trajectories are effectively produced by an intrinsic fitness landscape seemingly sharper under the deterministic sampling rule (i.e with less noise in the intrinsic fitness of a creature). It is important to note that these dynamics are clearer for higher mutation rates p_{mut} because they produce much more different parameters as mutations are the only way of introducing new parameters in the system. However, when looking at the relationship between the total number of parameters $|\mathscr{P}|$ and the mutation rates, as shown in figure 1.25, we can see that they relate sub-linearly. This indicates the presence of competitive dynamics in the system, since without competition the number of parameters will necessarily increase linearly with the mutation rate. Interestingly, after a rapid growth, the number of different parameters tends to stay very stable through time and this for all the different values of p_{mut} also indicating the presence of intrinsic regulation mechanisms.
When measuring evolutionary activity of the system, we also notice large differences when varying the mutation rates in the vanilla setting as shown in figure 1.26. Importantly, results show a rapid decline of evolutionary activity measures for greater mutation rates where the relation is best fitted by a power law with slope $\gamma = -0.5$ ($R^2 = 0.75$) for non-neutral activity and $\gamma = -0.71$ ($R^2 = 0.71$) for count-based activity. These results are consistent with results from Droop and Hickinbotham where they observed declines of evolutionary activity for too high mutation rates. When expressing the evolutionary activities as a function of time for different mutation rates, all of the curves are better fitted by a linear model. The slopes of these fitted curves are higher for low mutation rates and when plotting the slope of the curve over the mutation rate (figure 1.26) we can observe a decay following a power function.

When comparing the vanilla, dissipative and food models respective evolutionary activities we observe that dissipative and food models display significantly higher evolutionary activities for both count-based and non-neutral measures ($p < 10^{-5}$ for both, Mann-Whitney test). The dissipative model also shows higher evolutionary activity than the food model ($p < 10^{-5}$ for both, Mann-Whitney test). However, one should note that in the case of the food and dissipative models, total mass in the environment is not ensured to stay constant. This creates a bias in the evolutionary activity measures which takes mass into account. When removing the total mass effect by dividing the evolutionary activity by the total mass in the environment, we obtain

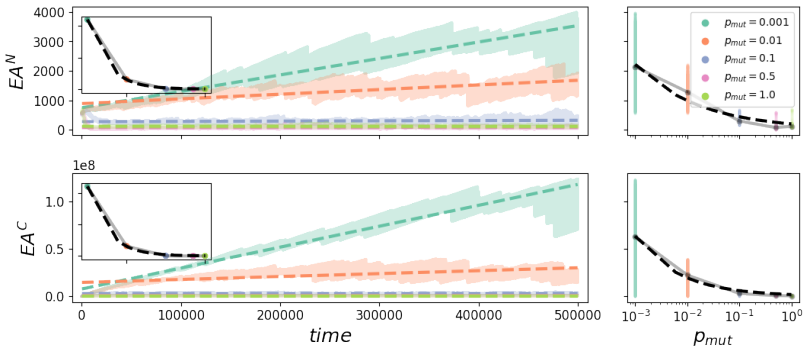


Figure 1.26: Evolutionary activity measures through time for different mutation rates p_{mut} in the vanilla model. Intuitively, the greater the total mass of a parameter p is, and the longer it survives, the higher will be its count-based activity ; with an additional penalization for stasis for the non-neutral activity (see section 1.3.4 for formal definitions). Measures of non-neutral (top) and count-based evolutionary activity (bottom) are shown. (left) Evolution of the evolutionary activity is shown as a function of time, dotted lines show the best fitting linear model. The inner plots show the the slopes of these models against their respective mutation rates p_{mut}, dashed lines indicate best fitting power function. (right) Average values of evolutionary activity over time are plotted against mutation rates, dashed line shows the best fitting power function.

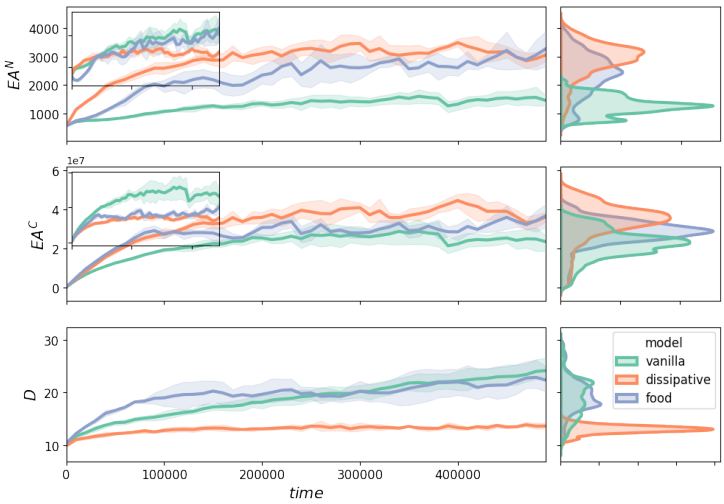


Figure 1.27: Comparison of non-neutral (top), count-based (middle) evolutionary activities and diversity (bottom) for the vanilla, dissipative and food models. (left) Metrics are plotted as a function of time for each model. For the evolutionary activity, inner plots shows the corrected measures (divided by total mass in the environment). (right) Distributions of respective metrics for each model (estimated through kernel density estimation).

the opposite relationships. We especially observe much lower measures in the dissipative case in comparison to the other two models. While the vanilla model shows a higher score than the food model when corrected activity is averaged over all time steps, we can see similar values in the end of simulations indicating a slower rise for the food model. Diversity measures (see figure 1.27, bottom) show striking differences where the dissipative model produces much less diversity than the other two. This might seem counter-intuitive as of the three it is the only one inputting new parameters. However, since the parameter map is never regularized, intrinsically evolving parameters can take any value which might be way out of the distribution from which we sample new parameters ($\mathcal{N}(0,1)$). The food model shows a quicker rise in diversity in comparison to the vanilla model but also stabilizes quicker while the vanilla model displays a steady constant linear growth of diversity.

1.3.6 Discussion

Due to the mass conservative nature of Flow Lenia, most of the patterns do not grow indefinitely into spatially global patterns (i.e patterns that diffuse on the entire grid, also called Turing-like patterns), therefore SLPs are much more common and easier to find. This is an important difference from the previous versions of Lenia, where one

needs to search or evolve for patterns that are both non-vanishing and non-exploding, and to constantly monitor their existential status (as it was the case in the previous contribution Section 1.2). Here, the mass conservation constraint acts as a regularizer on the kinds of patterns that can emerge.

Even though patterns generated by Flow Lenia are often static or slowly moving, we have been able to find creatures with complex dynamics from random search only which would be a difficult task in Lenia as most of the search space corresponds to either exploding or vanishing patterns. Furthermore, we have shown that the update rule parameters can be optimized with simple evolutionary strategies to generate patterns with specific properties and behaviors such as locomotion, chemotaxis and navigation through obstacles. Doing so in Lenia is a difficult task since the spatial localization of emergent patterns is not guaranteed necessitating more complex algorithms accounting for such a property.

Finally, we showed that the Flow Lenia system allowed for the integration of the update rule parameters within the CA dynamics, allowing for the coexistence of multiple update rules, and thus different creatures or species within the same simulation. The quantitative and qualitative analysis of trajectories of parameters through time, as well as the application of evolutionary activity metrics [280] and diversity metrics allowed us to shed light on the intrinsic evolution taking place in larger spatio-temporal scale simulations of this system. Intrinsic evolution, i.e evolution without externally defined stationary fitness function, is a particularly important feature of life as it supports its open-endedness through mechanisms such as niche construction.

We argue that such multispecies simulations represent an important step towards the design of emergent microcosms [293] in which could emerge intrinsic, maybe open-ended, evolutionary processes through inter-species interactions. Whereas environment design is poorly addressed and quite challenging in cellular automata systems, we believe that it is crucial to study the emergence of agency and cognition in those systems as argued in [294] and shown in previous contribution Section 1.2. By proposing different environment designs inspired by theories about origins of life and evolution, we have been able to study the evolutionary influences of such variations. By enabling the design of complex environmental features like inter-species interactions, walls, food or temperature, Flow-Lenia could represent a particularly interesting system to study theories on the origins of life or ecological theories of the evolution of complexity and cognition [41, 230].

Lot of exciting roads remain to be taken in order to fully capture the value of complex self-organized systems such as Flow-Lenia and explore their potential as models for studying theories about life, cognition and evolution. While we showed that evolutionary activity metrics are applicable to the Flow-Lenia system with minimal modification, we have shown that EA measures are able to characterize the effect of different experimental conditions (vanilla, dissipative and food) on the resulting evolutionary dynamics. In evolutionary biology and theories on the origins of life, the presence of a dissipative mechanism as well of limited shared resources are considered as key drivers of open-ended evolution in the natural world. However,

our results show an inverse tendency compared to these predictions, where the EA of the dissipative and food conditions are lower than in the vanilla systems. This is an interesting illustration of the interest of quantitative models such as Flow-Lenia, encouraging the formal definition of such mechanisms and measures. Our results show that common predictions on the origins of life actually strongly depend on their specific instantiation, since in our current model we actually observe the inverse tendency. However, we think that more improvements have to be made here. In particular, our working definition of species, i.e a specific point in the parameter space, might not fully capture what species are in our system. Looking at the distribution of parameters revealed the ubiquitous presence of clusters coherently moving in the high-dimensional space of parameters with occasional divisions (i.e. branchings). We believe that the correct definition of a species in Flow-Lenia might lie in these coherent clusters which might, through the scope of evolutionary activity metrics, shed new light on Flow-Lenia evolutionary dynamics. Another important contribution could also be to take into account the phenotypic outcomes of the parameters in the definition of species. Beside the notion of species, the definition of individuals could be reframed, or refined, as we often observe in simulations stable structures one would identify as an individual creature in the system but which are composed of elements defined by different sets of parameters. While evolutionary theories have for a long time thought about genes as the fundamental unit of selection, recent theories propose that the individual, or agent, should be seen as the fundamental unit of evolution [295]. Further study using Flow-Lenia might benefit investigating new methods for defining these individuals, for example using information theoretic measures of individuality [237]. Also, while measures of evolutionary activity represent a great tool for getting insights into the complex emerging dynamics of Flow-Lenia, other measures could be adapted and used in this setting. For instance, Patarroyo et al. proposed a framework based on assembly theory [297] aimed at quantifying the open-endedness of discrete CA [296].

In conclusion, we believe that Flow-Lenia represents an important step towards the realization of open-ended evolutionary dynamics *in silico*. By enabling great diversities of creatures to emerge, interact and evolve in complex large-scale environments, Flow-Lenia could lead to the emergence of complex cognition in the most as-it-could-be sense of the term. These emerging dynamics might shed new lights on studies about the origins of life and cognition.

1.4 Chapter conclusion

In this chapter, we provided in the first section Sec.1.2 a method based on gradient descent, diversity search and curriculum-driven exploration, allowing to easily learn the update rule of a CA leading to the systematic emergence of robust agents with sensorimotor capabilities. **These results shows how matter in an initially lifeless environment, whose dynamics is simply driven by the physic of the system, can self organize into artificial creatures displaying features of**

Summary

- Applying diversity search algorithms to a continuous cellular automaton enables the discovery of artificial creatures displaying features of sensorimotor agency with interesting generalization abilities.
- Introducing mass conversation in a continuous cellular automata enables multi-species simulations bootstrapping a proto-evolutionary mechanism.

simple "cognition". Remarkably, these macro-agents display coordinated behavior without a central control system, relying instead on the interaction of many atomic components. We also observed impressive generalization capabilities of the self-organized agents to conditions never seen during the search, reminiscent of the generalization abilities of self-organizing systems [51, 52], showing its potential usefulness for building robust AI agents.

In a second section, we introduced mass conservation into the Lenia cellular automaton. This enhancement led to more stable patterns and facilitated the incorporation of environmental elements and pressures. Most importantly, this extension enabled us to simulate "multi-species" interactions, where different sets of rules coexist within the same simulation. These multi-species simulations displayed competition between species for matter, ultimately leading to "intrinsic evolution". **This demonstrates how a lifeless environment, with only matter and local physic rules, can bootstrap evolutionary dynamics**. Our investigations into Lenia and Flow Lenia reveal promising avenues for understanding emergent complexity and eventually open-ended evolution with minimal engineered bias in self-organizing systems.

While the self-organized agents displayed promising robustness and generalization capabilities, their cognitive abilities still lag far behind those of agents within the mechanistic framework, which feature a clear body-brain distinction (such as reinforcement learning agents in AI). Further work could explore how to find rules leading to the self-organization of agents with more advanced forms of cognition such as memory and learning. In particular, we have seen that the mass conservation mechanism introduced in flow lenia tends to mostly produce spatially localized patterns, which were difficult to find in our first contribution Sec.1.2 (requiring to explicit this constraint in the engineered objective) due to Lenia' instability. In future work, this property of flow lenia is a promising direction to favor the discovery of spatially localized agents with more advanced forms of cognition, as exemplified in Sec.1.3.4.

In addition, work from the first contribution section.1.2, required an engineered objective function which had to be designed with expert knowledge. This contrasts with natural evolution which does not have an explicit target, yet still led to very complex beings. Similarly, Flow Lenia's open-ended evolution could potentially lead to the spontaneous (bottom-up) emergence of complex agent behaviors, including sensorimotor capabilities, purely through evolutionary dynamics rather than explicit optimization.

However, while emergent "evolutionary dynamics" in Flow Lenia is a promising avenue to achieve open-ended dynamics in silico, running flow-lenia simulation for a long time still seems to mostly lead to a "winner takes all" outcome where only one "species" covers the whole system. This suggests that there are still missing conditions in our simulations that could enable truly open-ended evolution, i.e. the continual evolution of creatures with increasingly diverse and complex phenotypes. Finding such conditions is our main long-term goal and we present below a few research directions we consider as promising.

For example, a potential next step is to explore the parameter space in Flow Lenia to search for system initialization and dynamics promoting sustained evolutionary activity. This could employ quality diversity search methods similar to those used in our first contribution Sec.1.2, though the challenge lies in developing robust metrics accounting for system increasing complexity.

Another avenue involves introducing engineered environmental elements into Flow Lenia, similar to approaches used in our first contribution Sec.1.2, to create niches that encourage diverse specialization. Ideally, such environment configuration would directly emerge through ecosystem evolution, rather than requiring explicit engineering. In fact, in this "all environment" simulation, interactions between agents and their environment dynamically reshape the environment configuration and could therefore create local niches through agents' activities and interactions.

In the next chapter, we will focus on studying such feedback loops, where agent adaptation results in changes of the environment properties, in turn modifying adaptation pressures on the agents.

Eco-evolutionary feedbacks and niche construction in multi-agent environments

2

In the previous chapter, we explored the emergence of individuals from an initial lifeless environment through the self-organization of simple entities (Sec.1.2). We further pushed this toward exploring the self-organization of open-ended evolution driven purely by the system's physics (Sec.1.3). In these simulations, the complex interactions between environmental entities created cascading changes in the environment's composition, generating novel selective pressures. In this chapter, we will now study in more detail this interplay between agents and the environment (notably their intertwined trajectories).

While our previous chapter adopted an enactivist framework (See Sec.1.2) – with no predefined interface between agents and environment and in which every component of an individual contributes to cognition -— we now shift to a more standard mechanistic framework (See Sec.1.2). This approach presupposes embodied agents distinctly separated from their environment, each equipped with a predefined set of sensors and actuators controlled by a centralized "brain". See Fig.2.1 for a comparison between the two frameworks. We also introduce pre-programmed adaptation mechanisms into our simulations. This transition from an enactivist to a mechanistic framework serves two main purposes. First, it allows us to implement agents that possess more sophisticated cognitive capabilities than those emerging in our cellular automaton simulations, as well as environments with more controlled dynamics. Second this shift enables us to study in more detail the interactions between adaptive agents themselves and their environment. In this chapter, we will focus in particular on studying how agents adapt to environmental constraints and opportunities, as well as on how this adaptation in turn modifies environmental dynamics.

As introduced in the beginning of this thesis Sec.0.3.2, in the natural world, "developing organisms are not solely products, but are also causes, of evolution" [204]: they actively shape their environment through niche construction, thereby influencing the fitness landscape, opportunities, and selective pressures [203, 205–208]. This interplay leads to eco-evolutionary feedbacks [204], where the evolved behaviors of organisms impact their surroundings, which in turn influence future evolutionary pressures, potentially shaping the behavior in an open-ended manner. (Fig.2.2).

In this chapter, we employ simulations lasting over an extended period of time with ecological inheritance [203] (the fact that the environment is also transmitted to the next generation Fig.2.2), allowing the study of the durable impact of the agent on the environment, the new pressures and opportunities it induces, and the overall eco-evolutionary feedback loop effects. This approach constitutes a significant departure from classical reinforcement learning and most evolutionary strategies tasks which typically employ short episodic

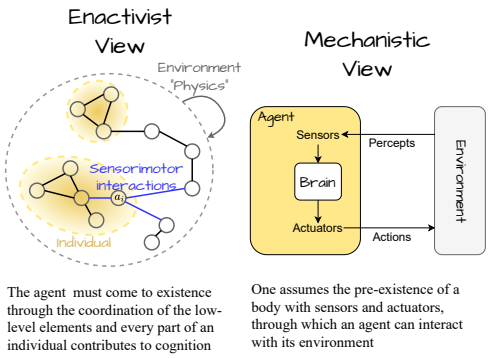


Figure 2.1: Comparison between the enactivist and mechanistic frameworks. More details in Sec.1.2.

Eco-evolutionary feedbacks

- "Organisms not only adapt to environments, but in part also construct them [202]. Hence, many of the sources of natural selection to which organisms are exposed exist partly as a consequence of the niche constructing activities of past and present generations of organisms."[203]
- "'Reciprocal causation' captures the idea that developing organisms are not solely products, but are also causes, of evolution." [204]

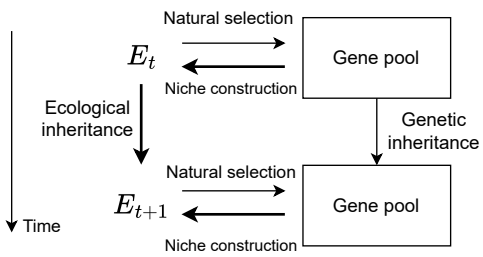


Figure 2.2: Feedback loop effects between agent and environment adaptation. Figure from [209].

training where the environment is regularly reset to an initial condition, preventing persisting changes and therefore feedback loop effects. Eco-evolutionary feedback loops can, in theory, from the non-stationarity they induce, lead to never-ending increases in complexity and diversity of the system (i.e. open-ended dynamics).

Common pool resources

Resource that is available to everyone (in the sense that it is hard to exclude people from using it) and where agents must collectively act to maintain it [298].

Collective niche construction often faces the challenge of common pool resources (CPR) dilemma. In CPR scenarios [298, 299], a resource is available to everyone (in the sense that it is hard to exclude people from using it) and agents must collectively act to maintain it and in particular not overconsume it. For instance, fishing in a lake presents a common-pool resource (CPR) challenge: while everyone has the right to fish, overfishing can lead to the depletion and potential collapse of the resource for the entire population. CPR environments are particularly interesting as a testbed to study eco-evolutionary feedback effects as agent consumption of the resource and their involvement in the maintenance of it directly affect the pressures and opportunities in the environment.

The CPR scenario is also inherently well suited to study the emergence of cooperation. However, this setting also involves a tension between collaborative management and individual incentives to maximize personal gain. In particular, CPR might be very prone to "free-riders" taking advantage of the good without participating in its maintenance, potentially leading to a "tragedy of the commons" [299] where the resource is depleted or is deteriorated due to agents greedily consuming it without maintaining it (e.g. overfishing). Additionally, the presence of CPR (i.e. a common resource to share) can lead to competition between groups, potentially escalating into an "arms race" to "fight" for this resource. This arms race can also contribute to the overall increase in complexity of the agents and the environment [215, 216]. We refer to Sec.2.1.2 for information on previous works exploring in silico experiments of CPR social dilemma.

CPR environments vary significantly in their maintenance requirements. Some only require agents to limit their consumption rate so the resource can regenerate, while others require a series of (collective) actions to be maintained. Furthermore, the potential benefits may scale with management sophistication: simple conservation strategies might yield modest returns, while advanced ecological engineering could unlock substantially greater resources.

The first section of this chapter examines a simple CPR setup where resource dynamics require only basic management skills to thrive. This allows us to focus on fundamental eco-evolutionary dynamics. Using a biologically plausible neuroevolution approach, we simulate hundreds of agents interacting within a large CPR environment for an extended period of time. Despite the environment's apparent simplicity, the combination of eco-evolutionary feedback and multi-agent interactions generates remarkably complex behavioral dynamics.

In the second section, we will then focus on CPR environments demanding a more sophisticated level of maintenance to study how agents collectively learn to eco-engineer their own environment advantageously. Using the emergence of agriculture as our primary case study, we design an environment where plants compete for resources

and agents can actively intervene to promote the growth of preferred species. This setup represents a more complex CPR scenario compared to the first section, as it requires agents to develop and execute multi-step interventions rather than simply managing consumption rates. This setup allows us to examine the environmental and agent-based conditions that favor the development of ecological engineering skills, while also analyzing their impact on population dynamics and social organization.

2.1 Eco-evolutionary Dynamics of Non-episodic Neuroevolution in Large Multi-agent Environments

Context

This contribution was done at FLOWERS in collaboration with Eleni Nisioti, previously postdoc in the team.

- Hamon*, G., Nisioti*, E., Moulin-Frier, C. (2023, July). *Eco-evolutionary dynamics of non-episodic neuroevolution in large multi-agent environments.* **In Proceedings of the Companion Conference on Genetic and Evolutionary Computation** (pp. 143-146).
  Paper, Code

I am co-first author of this article (Main contributor for experimental design, coding; data analysis and writing with Eleni).

A companion website associated with the publication, with videos, is also available at `https://sites.google.com/view/non-episodic-neuroevolution-in/`

This publication also led to the Master internship of Timothé Boulet to explore further this topic, as well as a collaboration with Max Taylor-Davies (School of Informatics, University of Edinburgh, Edinburgh, Scotland) that led to another paper (long oral at the International Conference on the Applications of Evolutionary Computation – evoApps – 2025) exploring the emergence of altruistic behavior akin to kin selection , more details in Sec.4.1.

Abstract

Neuroevolution (NE) has recently proven a competitive alternative to learning by gradient descent in reinforcement learning tasks. However, the majority of NE methods and associated simulation environments differ crucially from biological evolution: the environment is reset to initial conditions at the end of each generation, whereas natural environments are continuously modified by their inhabitants; agents reproduce based on their ability to maximize rewards within a population, while biological organisms reproduce and die based on internal physiological variables that depend on their resource consumption; simulation environments are primarily single-agent while the biological world is inherently multi-agent and evolves alongside the population. In this work, we present a method for continuously evolving adaptive agents without any environment or population reset. The environment is a large grid world with complex spatiotemporal resource generation, containing many agents that are each controlled by an evolvable recurrent neural network and locally reproduce based on their internal physiology. The entire system is implemented in JAX, allowing very fast simulation on a GPU. We show that NE can operate in an ecologically-valid non-episodic multi-agent setting, finding sustainable collective foraging strategies in the presence of a complex interplay between ecological and evolutionary dynamics.

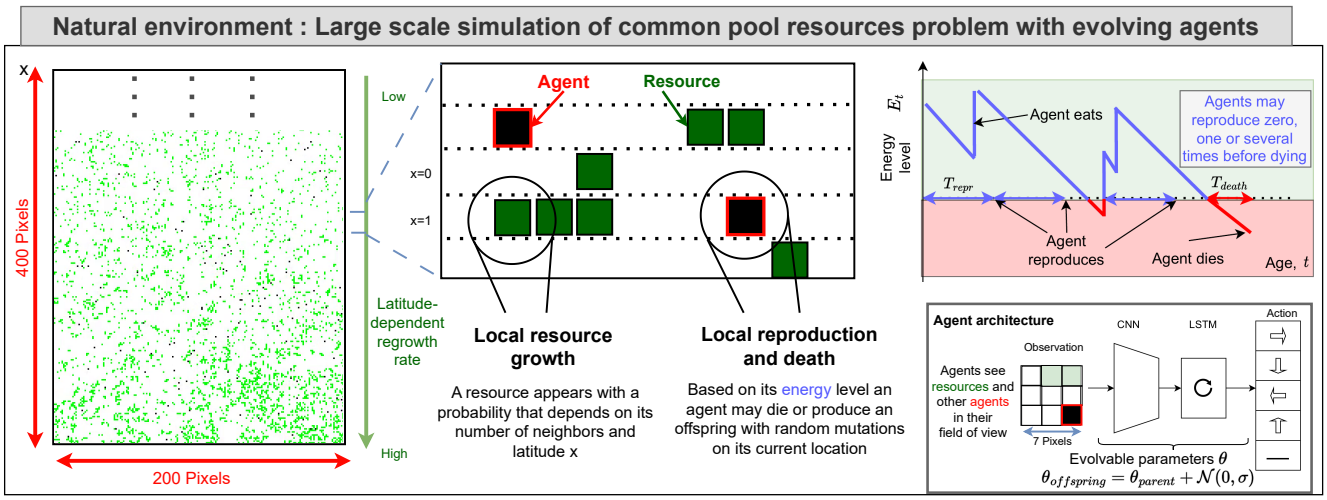


Figure 2.3: Our simulation environment (Left) is an extension of the Common Pool Resource (CPR) environment [300, 301] : a two-dimensional grid-world where some cells contain resources (in green) that the agents (in black) can collect. Resources grow depending on the presence of other resources around them (local growth, Middle) with an additional very sparse spontaneous growth, which means that over-consumption may lead to their local depletion. We introduce a latitudinal model of resource regrowth similar to [302] with higher resource regrowth rate in lower latitudes (bottom of the map) than in higher ones. We consider a very large environment of 200×400 pixel that can contain thousands of agents (in black). We prevent any environment and population reset during a whole simulation of $1.000.000$ time steps, enabling continual eco-evolutionary dynamics to take place. Each agent may reproduce or die according to a physiological model modulating its energy level as a function of life time and resource consumption (Top-Right). Agents reproduce according to a minimal criterion [303] of maintaining energy level above a certain threshold for a certain period of time. The population size therefore varies during the simulation according to the current amount of available resources and the current ability of agents to collect them. Each agent is controlled by a recurrent artificial neural network which takes as input limited local observations and outputs the navigation action (Bottom-Right). Evolution occurs through the mutation of a parent's network weights when it produces an offspring. We refer to section.2.1.3 for more details on the environment design.

2.1.1 Introduction

The main objective of this contribution is to propose a method for studying large-scale eco-evolutionary dynamics in agent-based simulations with a reasonable level of biological and ecological plausibility. For this aim, we implement a system with the following properties (see Fig. 2.3 for illustration).

Non-episodic simulation environment with complex intrinsic dynamics . We model our environment after common-pool resource (CPR) appropriation problems, where a group of agents competes for finite resources. We extend an existing environment of CPR appropriation [300] with the presence of multiple niches, where resources regrow proportionally to the density of nearby resources at different rates in different regions of the environment (Fig 2.3). We prevent any environment or population reset during a whole simulation run, enabling coupled environmental and population dynamics leading to complex eco-evolutionary feedback effects.

Continuous neuroevolution in a large, size-varying agent population The environment contains thousands of agents, each controlled by a neural network whose weights are optimized using neuroevolution [304]. Each network contains a memory component (LSTM), which enables adaptation within the agent's lifetime in the absence of weight updates. Thus the evolutionary process can be viewed as

an outer loop that optimizes the ability of agents to adapt to different environmental conditions.

Physiology-driven death and reproduction There is no notion of rewards, agents are instead equipped with a physiological system modulating their energy level according to the resources they consume, in a non-linear way. At the evolutionary scale, agents reproduce as long as they are able to maintain their energy level within a reasonable range and die if this level goes below a minimum threshold. This is departure from the notion of fitness-based selection and more in line with a minimal criterion selection [303]. Note that the population size can vary with time.

Natural vs. Lab environments: Evaluation methodology As we are interested in the system's ability to emerge interesting behaviors that hint to open-ended dynamics, evaluating it on pre-defined set of tasks would defeat our purpose. For this reason we have structured our simulation methodology as follows: we let the population of agents evolve for a long time in a single environment and study its behavior at a large global scale and at a smaller local scale. At the large scale, we study the dynamics of the system in what we call the "natural environment", i.e. the full simulation run, by monitoring population-wide and terrain-wide metrics. At the small scale, we first focus on local, interesting patterns of behaviors observed in the natural environment, such as individual agents that move in a consistent way or collective immigration and foraging patterns. We then form specific hypotheses about the potential drives of these behaviors and evaluate selected agents in specific "lab environments" that enable testing these hypotheses. These environments differ from the one used for evolving behaviors: they are much smaller and exhibit vastly different population and resource dynamics (we illustrate examples of such environments in Figure 2.4.E).

From the perspective of neuroevolution, our empirical study aims at answering the following questions: a) *can we realistically apply neuroevolution in multi-agent environments with thousands of agents?* b) *does a selection mechanism that allows agents to reproduce locally, without requiring generational resets, based on a minimal criterion suffice?* c) *does evolving networks in a multi-agent setting lead to the emergence of adaptation mechanisms?* From the perspective of multi-agent cooperation, our study targets the questions: a) *can we simulate systems with complex eco-evo dynamics where populations solving a CPR problem exhibit realistic behaviors?* b) *does evolving under a minimal criterion enable sustainability*? In the next section, we answer these questions in the affirmative.

Leveraging the GPU parallelization allowed by the JAX programming framework [289], we run large-scale continual simulations in grid-world environments with approximately $100K$ cells and thousands of agents (notably, a simulation of $1M$ time steps with such a population requires about only 20 minutes).

2.1.2 Background

Neuroevolution

Neuroevolution draws inspiration from natural evolution to create agents that learn to adapt through an evolutionary process rather than gradient-based optimization [304]. In a surprise to many, this simple process of selection and random mutations has recently performed competitively with the state-of-the-art in RL for playing Atari games [99, 102], and proven powerful in applications such as architecture search, where the non-differentiable nature of the search space prohibits gradient-based methods [100] and meta-learning, where the evolutionary process is conceived as an outer optimization loop that controls the intra-life learning plasticity of agents [305]. Multi-agent environments, which are particularly promising for neuroevolution as they naturally entail the concept of a population, have been identified as a frontier for this family of methods [306], arguably due to their computational complexity and challenging multi-agent learning dynamics .

Neuroevolution methods are classically performance-driven: solutions are selected based on their ability to solve a pre-determined task. Complexity-driven approaches, on the other hand, where solutions are chosen based on criteria not directly related to performance, such as novelty, have proven powerful in tasks for which the objective function is unknown to humans [307]. For a given criterion, neuroevolution methods can also differ on whether solutions survive only if they are ranked high within the population (survival of the fittest) or if their fitness is above a threshold (minimum criterion). The latter category is the least explored [303], but has the potential of preserving a larger phenotypic diversity within the population and is believed to be closer to biological evolution.

Finally, neuroevolution methods almost exclusively consider discrete, overlapping generations, at the beginning of which solutions experience mutation and selection simultaneously and the environment is reset to its initial conditions. We refer to this paradigm as episodic, borrowing terminology from RL, where recently it has been proposed to remove environmental resets, as they may introduce the need for human supervision [308] and are implausible from a biological perspective [212]. This setting, termed as non-episodic or continuous in RL, is harder to envision in evolution under survival-of-the-fittest, where dividing time into non-overlapping generations ensures that agents compete based on the same time budget.

Common-pool resource appropriation

CPR tasks abide in natural and human ecosystems: fisheries, grazing pastures and irrigation systems are examples of multi-agent systems where self-interested agents need to reach a sustainable resource appropriation strategy that does not exploit the finite resources. They

belong to a class of game-theoretic tasks termed as social dilemmas, which exhibit a tension between individual and collective motives: the optimal collective strategy is to forage sustainably but self-interested agents will cooperate only if others cooperate as well; otherwise they will consume resources until they deplete them, a situation called Tragedy of the Commons [299]. Ecological properties of these complex systems, such as the spatiotemporal variability of resources and organisms are believed to play a big part in shaping solutions to CPR problems [309]. From an ecological perspective, such settings give rise to scramble competition, where organisms of the same species appropriate resources at a rate contingent on their foraging ability, often leading to population bursts and crashes [310].

With recent advances in RL, computational studies of social dilemmas have managed to operate in simulation environments resembling the ones used in human lab studies, where agents can navigate a grid-world consuming resources [300, 301]. RL agents embody the self-interested trial-and-error learning paradigm and have confirmed our intuition that, when acting in a group, they cannot avoid a Tragedy of the Commons unless they employ some auxiliary mechanism for guarding against exploiters, such as learning to incur punishment [300] and reputation mechanisms [311]. These studies, however, remain far from approaching the complexity of real ecosystems, which may comprise thousands of organisms that do not necessarily follow the reward-maximization paradigm.

2.1.3 Methods

The environment

Our simulation environment is an extension of the CPR environment [300, 301] that the AI community has been using to study the emergence of cooperation in groups of self-interested agents: a two-dimensional grid-world where some cells contain resources (in green) that the agents (in red) can collect. Resources grow depending on the presence of other resources around them, which means that there is a positive feedback loop, with reduction in resources leading to further reductions. In addition to resources, the environment may contain walls (in blue) that kill agents trying to traverse them (see Figure 2.3 for an illustration of our environment).

At each time step t of the simulation a resource may grow in a pixel in a cell of the environment with location (x, y) based on the following three processes:

- a neighborhood-dependent probability $p_I(x, y)$ determines the probability of regrowth in a cell based on the number of resources in its neighborhood, I
- a niche-dependent scaling factor $c(x)$ is used to scale p_I. We employ a latitudinal niching model used in previous studies [302, 312]: the world is divided into N niches, each one having the form of a horizontal stripe of pixels so that a cell's location depends only on its vertical position x. We refer to $c(x)$ as the climate value of niche x.

- independently of its neighbors and niche, a resource grows with a constant low probability c

By modeling resource generation in this way we ensure that the resource distribution follows the CPR model, that it exhibits additional spatio-temporal variability due to the presence of niches and that resources do not disappear too easily, which can be problematic in reset-free environments. Thus, the combined regrowth rate for a resource r is:

$$p(x, y) = p_I(x, y) \cdot c(x) + c \tag{2.1}$$

A niche's climate value is determined by equation: $c(x) = (\alpha^x + 1)/(\alpha + 1)$, which returns values from 0 to 1 and allows us to control the relationship between niche location and climate to be from linear to exponential.

The agents

At each time step there is a variable number of agents K_t in the environment, each one characterized by its sensorimotor ability, cognitive capacity and physiology.

Sensorimotor ability An agent observes pixel values at each time step within its visual range (a square of size $[w_o, w_o]$ centered around the agent, as illustrated in Figure 2.3]). The pixel values contain information about the resources, other agents (including their number) and walls. At each time step an agent can choose to stay inactive or execute an action to navigate up, down, right or left.

Cognitive capacity An agent is equipped with an artificial neural network that outputs the action to undertake based on the current observation and whose weights are initialized randomly once at the start of the simulation. Its architecture (illustrated in Figure 2.3) is minimal: a convolutional neural network, an LSTM cell that equips the agents with memory by enabling policies conditioned on a trajectory of observatories and a linear layer that transforms hidden states to actions.

Physiology An agent is equipped with a simple physiological model modulating its level of energy: the agent is born with an initial energy value E_0 which, at every time step, experiences a linear decrease, and, if the agent consumes a resource, is increased by one (see Figure 2.3 for an illustrative example of how the energy level may change within the lifetime of a hypothetical agent). The energy is also clipped to a max value E_{max}.

Non-episodic neuroevolution

In neuroevolution (NE) a population of neural networks adapts its weights through random mutations and a selection mechanism that promotes well-performing policies. Under a classical NE paradigm training time is divided into generations, at the end of which agents reproduce to form the next generation [99, 304].

Our proposed system deviates from this paradigm in two respects:

- agents do not reproduce according to their fitness but according to a minimal criterion on their energy level;
- evolution is non-episodic: upon satisfying certain criteria an agent reproduces locally (the off-spring appears on the same cell as its parent), so that agents are added in an online fashion to the population, removing the need for a concept of generation.

Reproduction In order to reproduce an agent needs to maintain its energy level above a threshold E_{min} for at least T_{repr} time steps. Once this happens the agent produces an off-spring and is a candidate for reproduction again. Thus, agents may have a variable number of off-spring and do not die upon reproduction. We illustrate this relationship between energy level and reproduction in Figure 2.3. Reproduction is asexual: an agent's weights are mutated by adding noise sampled from $\mathcal{N}(0, \sigma)$

Death An agent dies once its energy level has been below a threshold E_{min} for at least T_{death} time-steps or if its age is bigger than a certain value L_{max}. Once this happens, the agent is removed from the population forever.

Evaluation methodology

The classical performance-driven evaluation paradigm in machine learning separates an experiment into two distinct phases: during a *training phase* the agents learn a policy and during an *evaluation phase* the agents act without learning in pre-determined tasks. In RL, these tasks were traditionally identical to the ones used in training, as RL agents were too brittle to generalize to unseen conditions [313]. Recent advances in meta-learning have enabled evaluation in a wide diversity of tasks, but require extensive training [184].

Evaluation in a complexity-driven paradigm is however more nuanced: as we are interested in the system's ability to emerge interesting behaviors that hint to open-ended dynamics, evaluating it on predefined set of tasks would defeat our purpose. For this reason we have structured our simulation methodology as follows: we let the population of agents evolve for a long time in a single environment and then study its behavior at a large scale, by monitoring population-wide and terrain-wide metrics and at a small scale, by focusing on local, interesting patterns of behaviors such as individual agents that

move in a consistent way or collective immigration and foraging patterns. We then form specific hypotheses about the potential drives of these behaviors and design environments that enable testing these hypotheses. These environments differ from the one used for learning behaviors: they are much smaller and exhibit vastly different population and resource dynamics (we illustrate examples of such environments on the right of Figure 2.3). This evaluation methodology should strike as familiar to the ALife and ecology communities and, we anticipate, will become more prevalent in AI studying open-ended skill acquisition. Borrowing terminology from ecology, we henceforth refer to the large-scale environment as a *natural environment* and the small-scale ones used for hypothesis-testing as *lab environments*.

2.1.4 Results

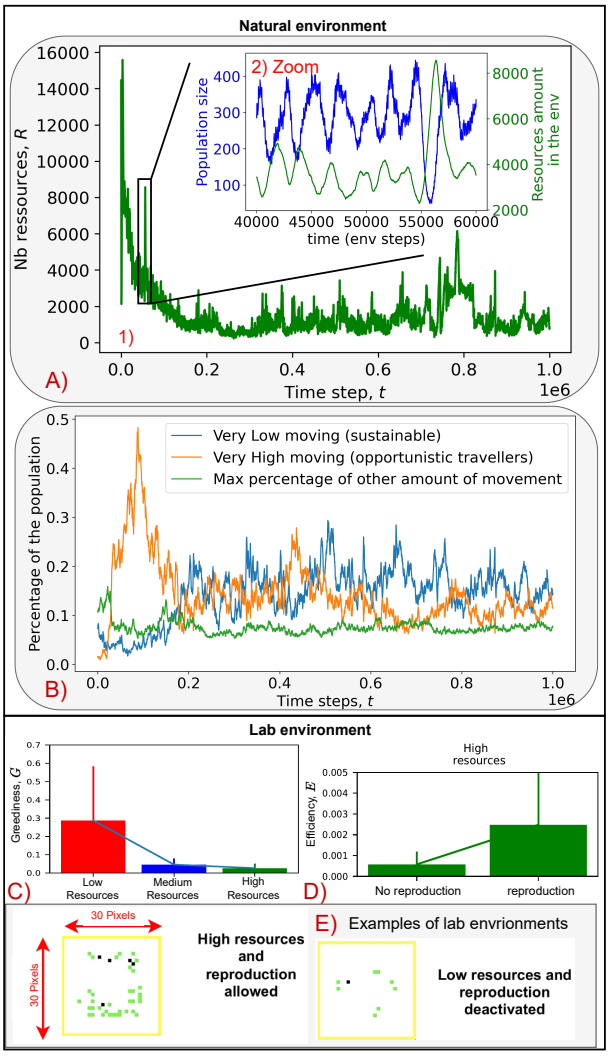


Figure 2.4: A) 1) Amount of resources in the environment over time and 2) zoom on a smaller timescale showing the interplay with population size (blue) B) Percentage of individuals with different amount of movement over time C) Greediness of a sustainable forager agent across evaluation environments that differ in the amount of resources. D) Average efficiency across the population in high resources task with reproduction activated and deactivated. Activating reproduction leads to increased resource consumption.

We will now study the evolution of a population in our proposed system and probe certain quantities during evolution. Note that this system required some tuning of the hyperparameters in order to find a stable environment, as exponential growth of both food and population can easily lead to collapse (and even did after several genera-

tions in 3 out of the 5 seeds launched). We will make a detailed analysis of one seed and refer to Appendix A.3.3 for an analysis of another one with a different eco-evolutionary path. We provide videos that show the real-time behavior of our system in a companion website (*https://sites.google.com/view/non-episodic-neuroevolution-in/*) as well as a repository containing code for reproducing our experiments (*https://github.com/flowersteam/EcoEvoJax*).

Details on the environment and hyperparameters characterizing the natural environment can be found in Appendix.A.3.1 and an explanation of how the metrics have been implemented and how statistical significance was tested for in Appendix A.3.2.

Eco-evolutionary dynamics

In this simulation, the evolution of the population is deeply interconnected with the evolution of resources. In Fig 2.4.A.2, we observe that at a small scale the population size (blue) and resources (green) present in the environment follow a predator-prey Lokta-Volterra dynamic [314]. These oscillations are interesting from an evolutionary perspective: easier phases with higher resources availability, in which the population grows and where diversity can emerge, are followed by high competition phases due to an increase in population and decrease in resources.

Coexistence of agents with different movement dynamics At the beginning of evolution (steps $0-200K$, starting with random agents), the environment has abundant resources which leads to to high-moving behaviors as an easy first strategy in this high-resource environment (Fig 2.4.A,B). Then, when the amount of resource decreases, we observe an increase in the number of low-moving individuals (Fig 2.4.A,B) exploiting local resource spots (from step $200K$). From this point, those two extreme strategies coexist in the agent population (Fig 2.4.B). This differs from previous related work in a similar environment [315], relying on a simpler agent architecture and a fitness-based reproduction condition, where only one strategy ended up populating the whole environment. Those extreme behaviors correspond to two distinct types of agents: high movement individuals are agents that have an "opportunistic traveler" strategy as they travel mostly in straight line but opportunistically exploit resource spots locally (especially from isolated resources from the sparse spontaneous growth) as soon as they see them. On the other hand, the low movement individuals exploit the spreading of resources by staying at the same interesting place (with resources around) and waiting for resources to spread. We qualify this waiting of resources as a sustainable strategy as agents do not consume resources greedily but rather keep these resources as a reliable source of respawn for more long-term survival (for themselves but also for their offspring that will inherit this place). We refer to video 1.a of the companion website for a visualization of these behaviors and to the next subsection for a more detailed and controlled analysis of the behavior (and diversity) of agents.

Evaluation in lab environments

How do the agents adapt their foraging behavior at an evolutionary and intra-life timescale to maximize their reproduction rate? In the natural environment, we saw that both population size and the spontaneous regrowth of resources may contribute to avoiding resource depletion. At an evolutionary scale, the population may adapt by regulating its size and updating its weights. But is it possible that the agents learned to adapt to different conditions they encounter in their lifetime in order to forage both efficiently and sustainably? This is the question the following simulations in the lab environments aim to address.

Does the density of resources affect agents' greediness?

Set-up There are three lab environments, with a single agent that cannot reproduce and resource regeneration deactivated, that differ in the amount of initial resources (see on Figure 2.4.E for an illustration of the low and high resources environments). In each of these lab environment, we measure the amount of greediness G, by dividing the simulation into non-overlapping fixed windows of 20 timesteps and checking in which of these windows the agent has at least one resource in its field of view (let's denote this number with T_r) and the number of these windows during which the agent consumed at least one resource (let's denote this number with C_r), so that $G = C_r/T_r$. We compute this measure on randomly sampled evolved agents from the end of the natural environment simulation. To quantify the effect of the density of resources we perform statistical tests comparing the greediness of each agent in the three tasks. More information can be found in Appendix A.3.2.

Our analysis showed that agents exhibit different qualitative behaviors that can be grouped in two types: a) agents for which no statistically significant differences appear between tasks. These agents correspond to the *opportunistic travelers* that we encountered in the natural environment and do not exhibit resource-dependent adaptation b) agents for which there are statistically significant differences between the low-resources and high-resources environment, with greediness in low-resource environments being higher. Overall, 9 out of the 50 agents exhibited this behavior (we illustrate greediness across tasks for one of these agents in Figure 2.4.C), which we refer to as *sustainable foragers*. These agents have learned to not over-consume resources when these are abundant, but stay close to them to consume them later and take advantage of the higher spread rate. On the other hand, low resources environment means slower spread (and may mean lattitude with lower regrow) which might exlain why even those sustainable agent prefer to take the resource and leave.

Does peer-pressure lead to greediness?

Set-up We use the high resources task but now allow agents to reproduce. This means that, after $T_{repr} = 20$ timesteps, new agents will appear, leading to competition for resources. Our hypothesis is that this will make agents more greedy. To test this, we measure efficiency E as the average amount of resources every individual consumes during the evaluation trial and average it across 50 agents and 10 trials. We then compare the difference in performance between the previous set-up (no-reproduction) with the current one, where we average across 50 agents and 10 trials to observe whether there is a population-wide effect.

As Figure 2.4.D illustrates, we observed a large change in the foraging efficiency of the agents when reproduction was on. Efficiency increased by a statistically significant amount, which indicates the sustainable foragers increased their greediness under peer pressure. However, we observed that, after an initial increase in resource consumption at the appearance of new agents, the group slows down again and its members tend to disperse and stay close to resources without consuming them (see companion website.B.1 for an illustration of this behavior).

2.1.5 Discussion

Our empirical study demonstrates that neuroevolution can operate in large multi-agent environments, lead to efficient behaviors even in the absence of episodic survival-of-the-fittest and help evolve agents that exhibit adaptation within their lifetime without requiring weight updates. Specifically in regard to the latter, we identified agents that change their policy depending on resource density and the presence of other agents. From an ecological perspective, our computational study proves that agents selected based on a minimal criterion learn sustainable behaviors and that the population exhibits dynamics that resemble those of natural populations, such as population size oscillations. We observed many interesting emerging examples of collective and individual adaptation, including:

i) Population size exhibits bursts and crashes that are correlated with the density of resources,
ii) The system goes through phases related to the sustainability of the agents' foraging behavior: resources and population size initially grow until over-population leads to near-extinction of resources which creates a drive for agents to forage sustainably,
iii) The sustainable population exhibits diversity in individual behaviors: some agents specialize in long-distance travel, opportunistically consuming resources they find on their way, while others forage locally, staying close to resources to take advantage of the spread of resources and consuming sporadically to avoid death,
iv) Agents' influence each others behavior: agents that forage sustainably when alone, temporarily increase their consumption when others enter their field of view and then revert back to consuming less.

Interestingly, points i) and ii) above could not be observed in a standard episodic training paradigm, where environment and population resets would prevent any eco-evolutionary feedback. In this respect, we are considering future experiments studying whether continual local reproduction, where offsprings are produced next to their parent, did enable some sort of kin selection –e.g. in the form of reducing parent's greediness as a way to favor the survival of their offsprings. Other future work could also focus on studying to what extent the memory component of the agent's cognitive architecture contributes to intra-life adaptation.

In the past, ecologists have hinted at the limitations of an anthropocentric view on intelligence [316]: if we search for intelligence by looking at performance metrics only in tasks that we excel at, then we will inevitably miss a big part of the natural kingdom. Our study hints at a similar conclusion for artificial agents: evolving agents in natural environments with complex spatiotemporal dynamics in the absence of rewards and examining their behavior in toy lab environments may bring us closer to our quest for open-end behavior in artificial systems.

2.2 Discovering agriculture through multi-agent reinforcement learning

Context

This contribution is the result of a collaboration with Ricard Solé (Complex system lab, Universitat Pompeu Fabra, Barcelona, Spain) and Martí Sànchez-Fibla (CSIC, Universitat Pomepu Fabra, Barcelona, Spain). In particular I did a 3 months visit in the Complex system lab in Barcelona in 2024.

This work is still a work in progress and the results of this section are still preliminary.

The discovery of agriculture is often seen as a major behavioral transition in the human species, paving the way toward modern technological development, increase in population size, and large-scale social organization [317]. However, the human species is not the only one having discovered agricultural practices: it is also the case of some ant species, e.g. fungus farming [318]. Insect fungiculture and human farming share common fundamental traits that are characteristic of advanced forms of agriculture [319]: a) Frequent seed planting, b) Dedicated cultivation of the crop in different forms: soil fertilization, protection against herbivores/fungivores, parasites, or diseases, c) organized harvesting of the crop, d) mutual nutritional dependency on the crop.

In this contribution, we study the environmental and cognitive factors promoting the emergence of agricultural practices in populations of artificial agents. We consider a simulated grid world with three plant species that compete with each other, with plant growth influenced by the presence of a fertilizer. We place a population of reinforcement learning agents in this environment, which are rewarded differently when they consume one plant species or the other. We experimentally control both cognitive factors (e.g. reward discount factor and exploration bias) and environmental factors (e.g. plant growth and inter-plant competition parameters) in different simulation conditions. We analyze the effect of these factors on the ability of agents to discover agricultural practices in a multi-agent setting. We find that under the assumptions of our model, the discovery of agriculture is favored by 1) a cognitive architecture favoring exploration and long-term returns and 2) the scarcity of naturally grown resources. We then explore in more detail the collective behavior of the agents that learned agricultural practices, for instance showing division of labor.

2.2.1 Simulation details

Environment dynamics. The environment is a gridworld with 3 types of plants:

- A **"yellow plant"** beneficial to the agents. This plant propagates through seed propagation (see below).

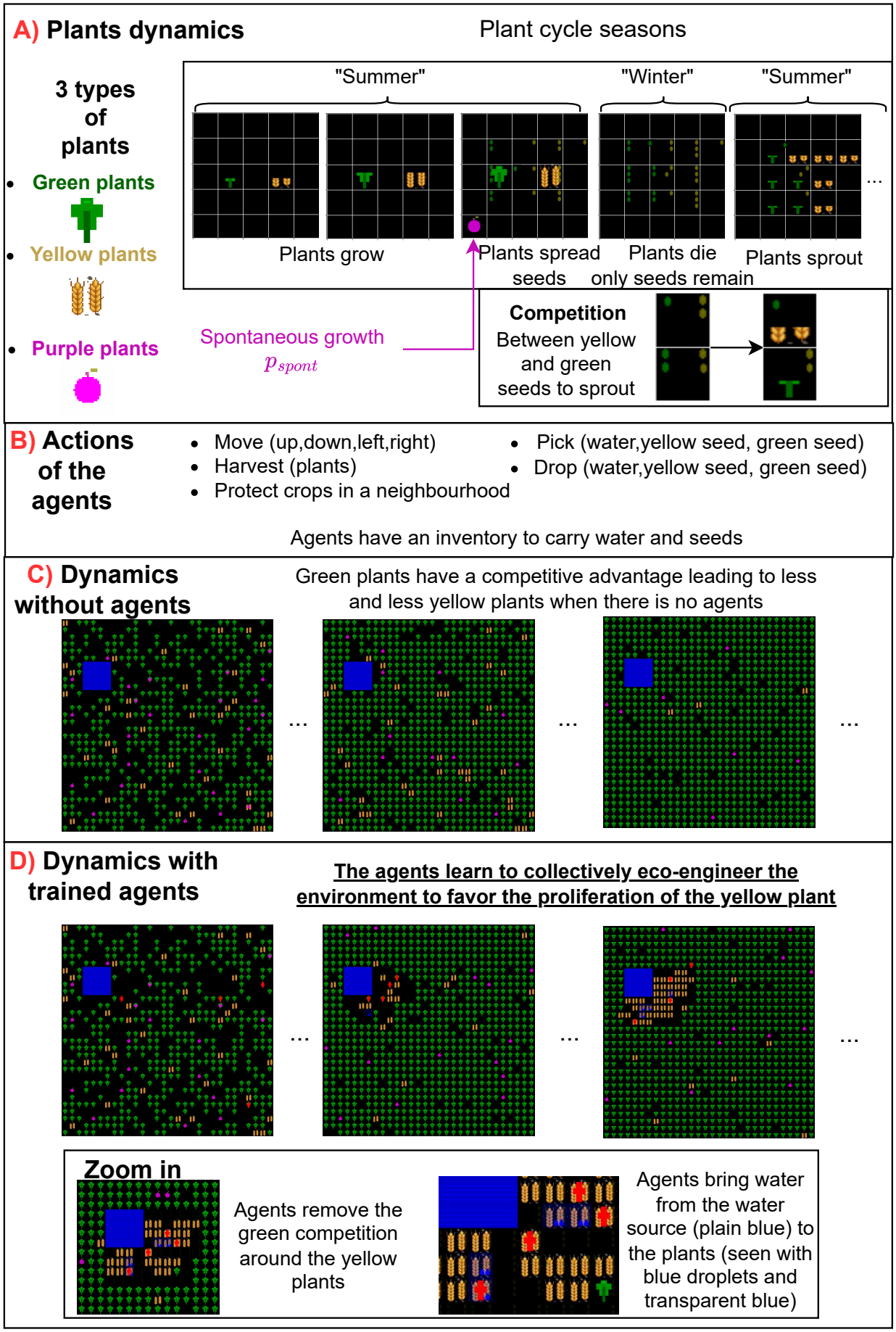


Figure 2.5: Overview of the system dynamics. A) plants dynamics. B) List of the agents' actions. C) Dynamics of the environment without agents, the yellow plants progressively disappear from the competition with the green plants. D) . We don't display seed in C) and D) for readability.

- A **"green plant"** competing with the yellow plant and useless for the agents. This plant also propagates through seed propagation.
- A **"purple plant"** that is beneficial to the agent, but does not spread and rather grow spontaneously (the reason for this difference is given below).

The plants follow seasonal cycles where plants grow during a "summer" season and die at the beginning of a "winter" season as depicted in Fig.2.5.A . More precisely:

- **Seed spreading. Yellow and Green plants** grow during the "summer season". A few steps before the end of the season, they spread seeds to the neighboring cells. At the beginning of the "winter season", the plants die and only seeds remain. At the beginning of the following summer season seeds have a probability to germinate to begin a new plant cycle. The probability of a seed to germinate depends on the number of seeds on this cells, including seeds of the other color plants.
  In fact, yellow and green plants are in competition, with an small advantage toward green plants (Fig.2.5.A). This allows to test the ability of the agents to eco-engineer the environment to favor the yellow plant growth by altering its competitor.
  Seeds that did not sprout at the beginning of a new season stay in the grid to sprout later but have a probability to disappear.
  These two plants induce a Common Pool Resource (CPR) appropriation scenario (Sec.2.1.2). In particular, as depicted in fig.2.5.C-D, without any maintenance from the agents, the yellow plant slowly decays in the grid over the episode until nearly disappearing. The agents can also overconsume it, accelerating this decay.
  Therefore, to maximally benefit from the yellow plant, the agents have to not overconsume it but can also favor its spread through different actions.
  The CPR nature of the system also expose the group of agents to free-riders – agents taking advantage of the resources without participating in their maintenance.
- **Spontaneous growth. Purple plants** have a probability p_{spont} to appear spontenously on a cell at each timestep during the "summer season". The purple plants disappear during the winter season.
  The dynamic of the purple plant differs from the two others as it relies on spontaneous grow (instead of seed spread). The reason of this design choice is to introduce a simple purple-plant foraging strategy that can be exploited by the RL agents. Indeed, the purple plant does not require any maintenance but spawns randomly at a rate p_{spont}. With this parameter we can therefore experimentally control the gap in returns between foraging and agricultural strategies and study how environmental and cognitive factors favor one or the other.

An episode consists of several seasonal cycles. In the following experiments, the period of the seasons is 40 (half summer half winter) with a total number of timesteps of 1024, i.e. more than 25 seasons.

The environment also contains **sources of water** (blue in Fig.2.5.D) that the agent can use to water the soil. Water acts as a fertilizer: plants that are in the 3x3 neighbourhood of a watered cell give more reward when harvested. Water on soil evaporates and, therefore, has a probability at each timestep to disappear. Agents, therefore, have an interest in constantly bringing new water to the soil.

We provide in the appendix.A.4.1 more details on the dynamics of the environment.

Agents dynamics. The simulations are conducted in a multi-agent scenario with 4 agents in the grid. The agents possess an inventory that allows them to carry seeds and water (without limit in the amount).

Each agent can execute the following actions:

- **Movement**: up, down, left, right.
- **Pick** (water,yellow seed,green seed). Only one seed can be picked at a time.
- **Drop** (water,yellow seed, green seed). Only one seed can be dropped at a time.
- **Harvest** (yellow plant, green plant, purple "foraging" plant).
- **Protect action**. This action prevents any agent from removing the yellow plant in a 3x3 local neighbourdhood of the agent that performed this action (only for the timestep this action is performed). This type of action was introduced in Perolat et al. paper on multi-agent RL in a common pool resources problem [300], where it was necessary to learn a collective sustainable behavior.

Some of the actions, such as dropping and harvesting, have a small cost in order to promote "intelligent" use of them and prevent abuse. More details in the appendix A.4.2.

Reinforcement learning training. In this work, we use episodic reinforcement learning (RL) to efficiently train the agents. However, we still allow long episodes of 1024 steps to allow for the long-term eco-engineering of the environment.

In RL, an agent observes an environment and performs actions on it that incur rewards, aiming at maximizing the rewards it accumulates. At each time step t of an episode that lasts for T time steps the agent (partially) observes the environmental state s_t, performs action a_t and receives reward r_t. The partial observation of the agent is a function of the environmental state: $o_t = obs(s_t)$, for instance corresponding to the agent's local neighbourhood. The policy $\pi(a_t|o_t)$, which describes the agent's behavior as a mapping from agent's observations to actions, is interactively learned from experience by maximizing the cumulative reward $G_T = \sum_{t=0}^{T} \gamma^t r_t$, where γ is a parameter quantifying how heavily future rewards are discounted [108]. Small γ results in nearsighted policies that mainly favor immediate reward while large γ favors long-term policies.

In this contribution, the training is decentralized: each agent learns independently using proximal policy optimization (PPO)[320] on its own history of interactions with the environment. This means that agents do not have access to other agents' observations, actions, and rewards.

To encourage continuous exploration, the PPO objective function incorporates an entropy term that nudges the policy distribution toward a more uniform spread over possible actions. This ensures that all actions remain under consideration, mitigating premature convergence to suboptimal strategies. The influence of this entropy term is regulated by the parameter λ_{entr}, which effectively adjusts the degree of exploratory behavior.

Each agent's action policy is a trainable transformer-based neural network. It takes as input the history over a time window of: the agent's local observation of the environment in a 11×11 neighbourhood around it, the state of its inventory, and the time of the season. Based on these inputs, it outputs the actions. We refer to appendix.A.4.2 for more details on the observation space and architecture of the agents.

We use our open-source code implementation of transformer-based agents with PPO, more information in Sec.4.3.

2.2.2 Measures

We report in this section the different measures used to track our simulation dynamics and the emergence of agriculture.

- **"yellow grown"** measures the total number of yellow plants that sprouted in the environment. This is measure is useful to track how much the agents participate in the proliferation of the yellow plant.
- **"Drop water"** measures how much the agents water the soil in the environment.
- **"Seed planted"** measures the amount of yellow seeds actively by planted the agents in the environment.
- **"Protect action"** measures the number of times agents use the protect action to protect crops from other agents.
- **"Agent neihbourhood"** measures the average number of agents in a close neighbourdhood of each agent. This measures indicates how much agents group together.
- **"Movement"** measures the average number of cells traveled by the agents over a time window. This is useful to track how much the agents move to find resources versus stay at the same place to potentially eco-engineer.
- **"Forage plant eaten"** measures the amount of purple plants consumed by agents.
- **Sustainability** (taken from[300]) measures the average time at which the agents consume resources. A low value means that the agents consume much more at the beginning of the episode (potentially hinting towards over consumption), while a high value means that the agents consume more at the end of the episode (hinting towards sustainable behavior).

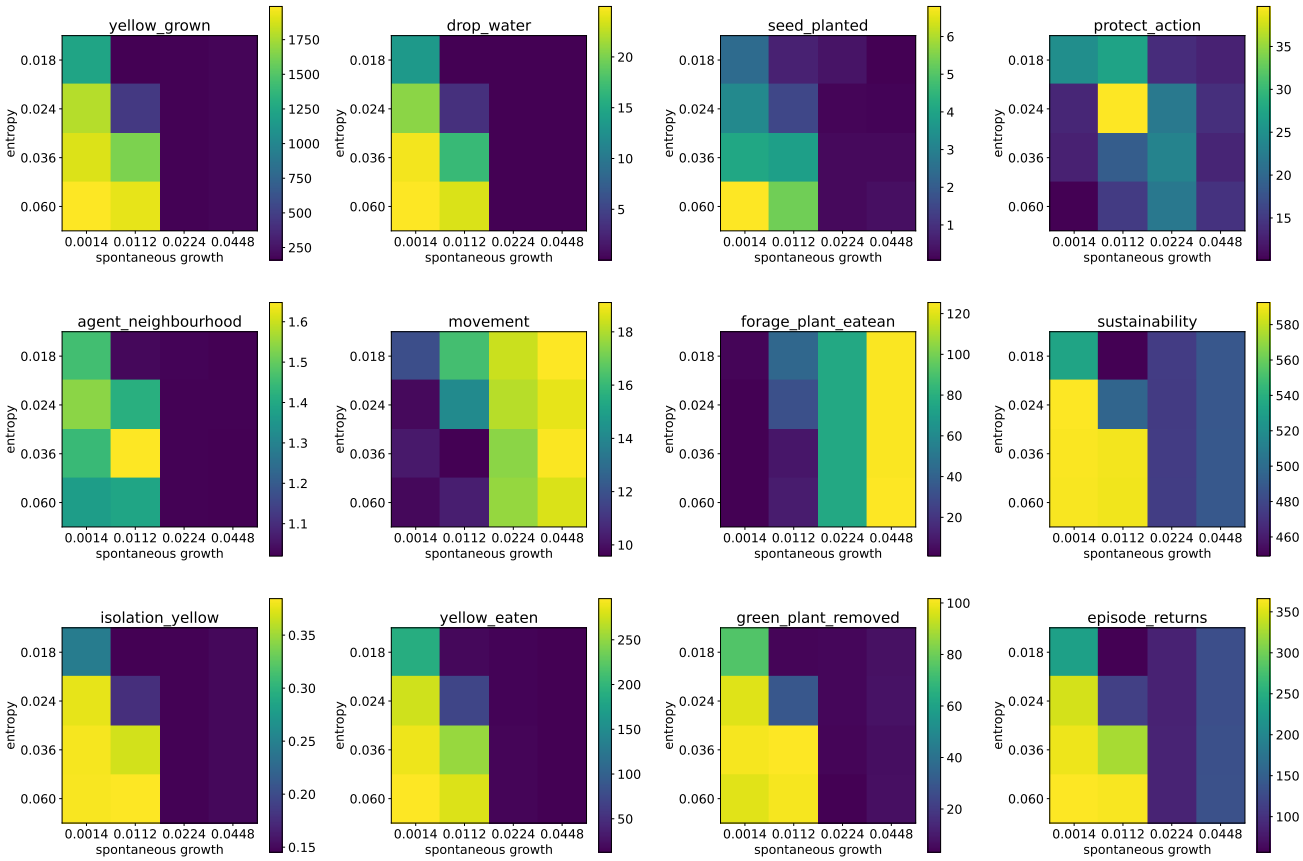


Figure 2.6: Heatmap parameter analysis over the spontaneous growth probability of the purple plant p_{spont} and the entropy bonus term (favoring exploration) λ_{entr}. We observe a sharp transition from measures indicating foraging strategies (high p_{spont} and low λ_{entr}; top right part) to measures indicating agricultural practices (low p_{spont} and high λ_{entr}; bottom left part). Each parameter couple is tested over 3 seeds, we report the average value. The experiments were performed with $\gamma = 0.999$

- **"Isolation yellow"** measures how much yellow plants are isolated from the green plants. A high value means that there is on average very few green plants in the direct 3x3 neighbourhood of the yellow plants.
- **"yellow eaten"** measures the amount of yellow plant eaten by the agents.
- **"green plant removed"** measures the number of green plants that were removed by the agents. This is a proxy for how much the agents remove the competition from the yellow plant.
- **"Episode returns"** measures the total reward obtained in average by agents over an episode.

2.2.3 Preliminary results

What are the roles of environmental and cognitive factors in favoring the discovery of agriculture?

We report in Fig.2.6 a parameter analysis over:

- **An environmental parameter:** the spontaneous growth probability p_{spont} of the purple plants. The higher this parameter, the

more abundant the purple plant is in the environment. It therefore indirectly controls the potential total return of a strategy consisting in simply foraging the purple plant. We predict that low values of p_{spont} (i.e. scarcity of the purple plant) should favor the discovery of an agricultural strategy.

- **A cognitive parameter:** the entropy bonus term in PPO λ_{entr}. This parameter is a proxy for the incentive of the agent to explore during learning.

For this analysis, the discount factor γ is fixed to 0.999. This high discount factor favors the learning of action policies that take into account cumulative reward over the long term.

We observe in Fig.2.6 a sharp transition in nearly all measures in the same region. In particular, the bottom left region, of higher entropy (exploration) bonus and low spontaneous growth shows measures indicating the emergence of agriculture. Indeed, we observe in this region that the amount of yellow plants grown is much higher, that the agents plant many more seeds, drop much more water, and remove many more green competitor plants. We also observe a higher sustainability metric, less movement, and less 'foraging' of the purple plants. The fact that agents move less and eat far fewer foraging plants indicates that the agents stay at the same place, focusing on consuming the yellow plants they are helping to grow. We provide in the following section 2.2.3 a deeper analysis of the behavior of the agents and the learning dynamic.

Interestingly, agriculture does not emerge across a significant portion of the parameter space, despite offering a much higher total reward (Fig. 2.6). Indeed, the spontaneous growth of the purple plant has no influence on the environment agricultural potential. Consequently, even in the high p_{spont} region of the parameter space, greater rewards could be achieved by adopting agriculture. However, as p_{spont} increases, foraging becomes an increasingly entrenched local optimum, making it even more difficult to transition to agriculture.

We observe similar results with the discount factor γ – specifying how much the agent takes into account future rewards in its choice of action – as shown in Fig.2.7 (See appendix.A.4.3 for the heatmap with γ).

From these results, we conclude that, under the assumptions of our model, the discovery of agriculture is favored by an environmental and two cognitive factors. First, agriculture is hardly discovered in environments with an abundant access to foraging resources, even though it could still yield higher rewards if it were discovered. Our interpretation is that, in such conditions, a foraging strategy constitutes a strong local optima from which it is then hard to escape. Second, the discovery of agriculture requires a cognitive ability to make decisions based on long-term predictions of future outcomes and the incentive to explore. The importance of gamma is expected as, with low discount factors, agents are near-sighted while eco-engineering requires an immediate cost (in terms of spending time to eco-engineer and not consuming the resources) for a future better environment with rewards.

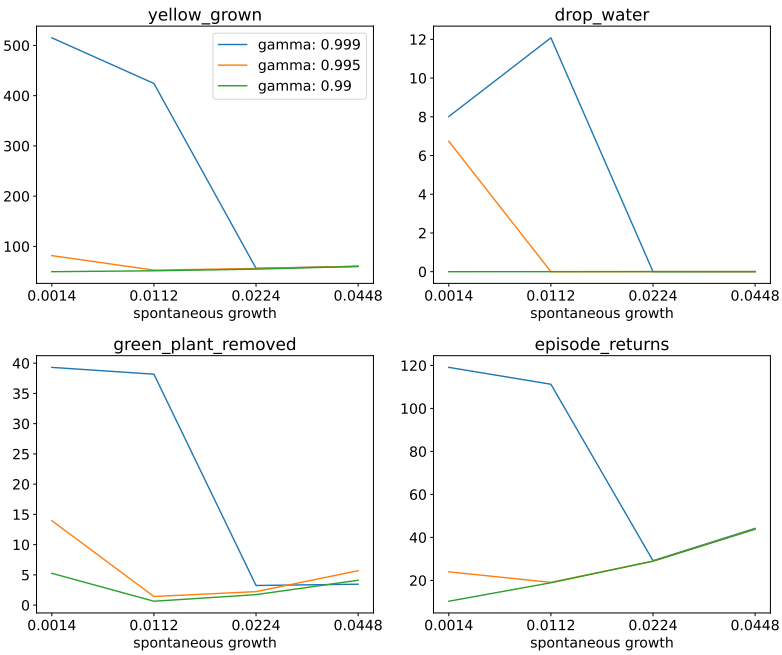


Figure 2.7: Parameter analysis of p_{spont} and γ. We observe a sharp transition in the metrics indicating agricultural practices as well as the episode returns. Agriculture only emerges fully for a very high value of γ and a low value of p_{spont}. Each parameter couple is tested over 3 seeds, we report the average value. We can clearly see that the episode return (bottom right) is much higher with agriculture yet does not emerge for high value of p_{spont}. The experiment were performed with $\lambda_{entr} = 0.036$

In favorable conditions, what is the learning dynamics of an agent population discovering agriculture?

We now choose parameters favoring the emergence of agriculture in the previous experiments and study in more detail the resulting learning dynamic.

Qualitative analysis of the agriculture behavior. As displayed in Fig.2.5.D, agents learn to make their "field" close to the water source, supposedly for two reasons: First, it allows them to get water easily; second, the water source allows having one side less of pressure from the green plant. We also observe (Fig.2.5.D) that the group of agents effectively learns to make a controlled field of yellow plants clearly separated from the green plants – mitigating its competitive pressure on the field.

Learning dynamics. We report along training time, the metrics described above (Sec.2.2.2), for one seed of training leading to agriculture (Fig.2.8, $\gamma = 0.999, \lambda_{entr} = 0.036, p_{spont} = 0.0112$). We observe a progressive steady increase in the yellow plant growth metric, indicating that the agents learn to favor its growth. At the same time, the amount of "foraging eaten" decreases, as well as the amount of movement, indicating that the agents abandon foraging at the same time. We also observe that the agents start to drop water very early and then follow a similar increase as the yellow growth metric.

Several other measures have a seemingly correlated dynamic along training. Following the first phase of the metrics described in the previous paragraph, we observe at the beginning of learning an increase in the grouping of agents (seen with the metric "agent neighbourhood"), which seems to correlate with an increase in the use of

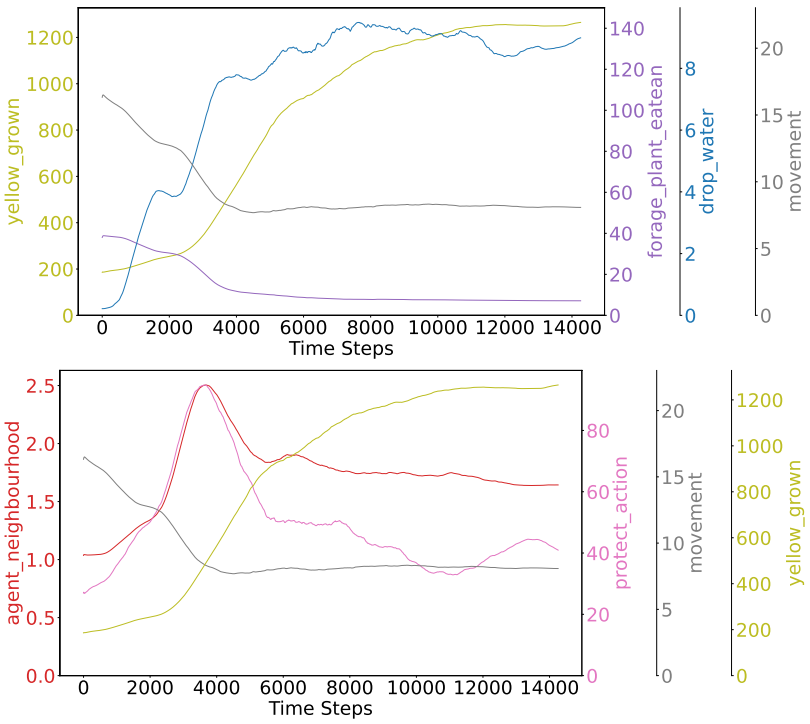


Figure 2.8: **Learning dynamics.** We report different metrics (Sec.2.2.2) along training time (x axis), for a simulation displaying the emergence of agriculture. We observe progressive steady increase in several eco-engineering metrics such as the growth of the yellow plants, and the dropping of water. This trend is acompanied by a decrease in movement and consumption of the foraging purple plant indicating agents abandoning foraging. We also observe a first increase in the grouping of agent ("agent neihbourhood") correlated with an increase in the use of the protect action, which then both decrease.

the protect action. Similarly, as the grouping of agents decreases slightly (potentially as they learn to collectively eco-engineer in a better way and cover a larger territory), we also measure a decrease in the amount of protect action used.

Further works could further explore, through more measures, those learning dynamics and in particular the several phases we observe.

Specialization and division of labor. We report in figure.2.9 measures per agent at the end of training for a specific training run ($\gamma = 0.999, \lambda_{entr} = 0.036, p_{spont} = 0.0112$).

Interestingly, we observe specialization of the agents and division of labor: the agent 3 (red bars) is the only one bringing water to the crops while the agent 1(blue color) is the main contributor to planting seeds (and protecting) but participates less in removing the green plants.

Such results highlight the interestingness of multi-agent decentralized training enabling agents to specialize, potentially leading to a collective division of labor. Division of labor is in fact a central concept across Szathmáry's framework of Major Transitions [321].

Note however, that most simulations where agriculture emerged did not have such a clear specialization, most of the time the agents participated in mostly all tasks (dropping water, bringing seeds, removing green plants and potentially protecting the crops from free riders). Here again, a more thorough analysis will be required.

We refer to appendix.A.4.3 for specialization plots for other environmental and cognitive parameters.

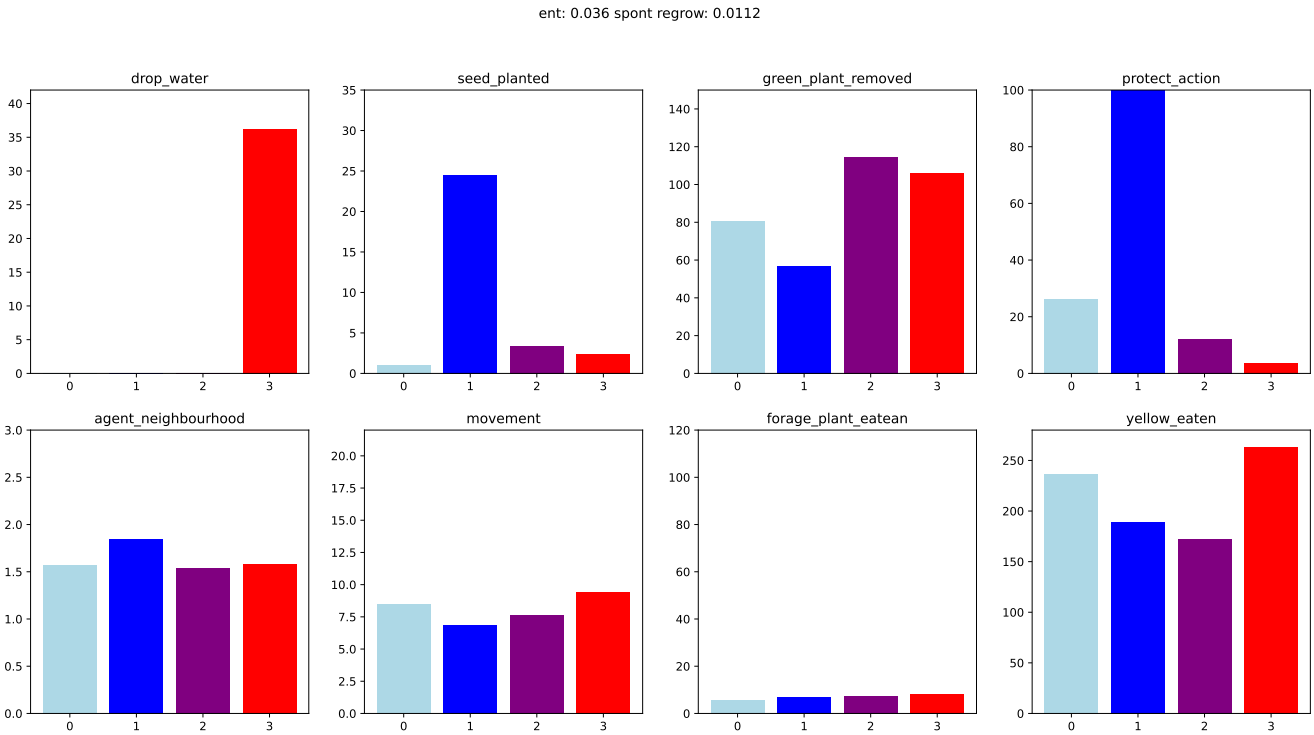


Figure 2.9: **Specialization of agents.** We report metrics per individual. Each color corresponds to a different agent. We observe specialization and division of labor. For example red agents drops water while the blue one plant seed and protect but remove less green plants than the others. The parameters used are : $(\gamma = 0.999, \lambda_{entr} = 0.036, p_{spont} = 0.0112)$

Interestingly, in the region of the parameter space where agriculture emerges, we observe in Fig.2.9 that every agent participates in the eco-engineering to some extent. In particular, we don't observe strong free-riding behaviors, in the sense that all agents contribute to eco-engineering activities, either through dropping water, planting seeds, or removing green plants. Further work could perform ablations (or parameter analysis) – for example, on the protect action – to explore the reason for this absence of free-riders.

2.2.4 Conclusion

In this contribution, we introduced an environment to explore the emergence of agricultural practices in groups of learning agents. We show in preliminary results that groups of agents trained with decentralized reinforcement learning are able, with the right environmental and cognitive parameters, to learn cooperative eco-engineering to favor the growth of a beneficial plant.

Our preliminary parameter analysis reveals that:

- Agriculture is hardly discovered in environments with an abundant access to foraging resources, even though agriculture could still yield higher rewards if it were discovered. Our interpretation is that, in such conditions, a foraging strategy constitutes a strong local optima from which it is then hard to escape.

- The discovery of agriculture requires a cognitive ability to make decisions based on long-term predictions of future outcomes and the incentive to explore.

This parameter analysis is, however, still limited and would benefit from testing more values, as well as other parameters. However, such parameter analysis can become rapidly computationally costly.

This contribution would also benefit from a deeper behavior and learning dynamic analysis than the preliminary results provided in Sec.2.2.3. Further work could also analyze in more detail the conditions favoring the emergence of specialization that our study revealed.

The emergence of agriculture is closely tied to social organization, particularly population size and growth [322, 323]. While the preliminary results presented here involve only four agents, future research could explore larger populations, including dynamically expanding groups. Promising findings, detailed in Appendix.A.4.3, indicate that in such environments, sustained population growth can occur without collapse, as new individuals contribute to resource production by joining the workforce.

Another hypothesis in human behavior ecology is the importance of storage in the organization of society and the emergence of agriculture [324, 325]. Further work could introduce the need to store resources, for example, during winter, to further test these hypotheses.

2.3 Chapter conclusion

In this chapter, we explored the feedback loop effects between adaptive agents and the environment, potentially leading to complex environmental trajectories and agent behaviors.

In the first contribution Section.2.1, we showed the important effects of eco-evolutionary feedback in large-scale multi-agent experiments with hundreds of agents despite a simple environmental dynamic. This work also showed that neuroevolution was capable of evolving efficient sustainable behavior in this complex scenario through physiological reproduction only, without any explicit objective or fitness function being maximized.

In particular, we observed the evolution of complex niche construction, which we explored further in the second section, exploring the emergence of agriculture, where agents' actions leveraged the environment's capacities to provide more abundant and sustainable resources.

Our findings in Section.2.1 revealed the emergence of sustainable behaviors among agents, which we hypothesize were evolutionarily selected due to their role in enhancing offspring survival over several generations through resource preservation. Building on these observations, our subsequent work which we briefly describe in Section.4.1 examines this phenomenon more systematically through dedicated simulations focusing on the kin selection of feeding behaviors. These

Quick summary chapter 2

- Large scale experiments showing the important effects of eco-evolutionary feedbacks.
- Neuroevolution of efficient sustainable behavior through physiological reproduction, without any explicit objective being maximized.
- Different behavioral strategy coexisting, elicited by isolation and behavioral tests in "lab environments".
- Learning of collective eco-engineering strategies with the emergence of agriculture.
- Eliciting the conditions favoring the discovery of agriculture.

investigations demonstrated how evolutionary pressures can favor the development of altruistic behaviors toward offspring.

In addition, the continual nature of our simulations, where agents coexist with their offspring, also creates conditions conducive to cultural transmission. While several works [150–152] have demonstrated passive teaching—where learners observe agents performing tasks—future research could investigate the emergence and kin selection of active teaching behaviors.

In fact, while we intentionally employed simplified environments to isolate and analyze eco-evolutionary feedback loops in the first section, future work could incorporate additional complexity for more complex agent-environment co-adaptation. This expansion could occur along two complementary axes: agent capabilities and environmental dynamics. For agents, the introduction of explicit communication mechanisms could facilitate the emergence of sophisticated cooperative behaviors and potential teaching. Environmental complexity could be enhanced by incorporating elements from our agricultural studies (Section.2.2) or introducing compositional dynamics – the possibility to produce new elements in the environment (such as tools), or to change the properties of existing ones, by composing other elements. The latter is particularly promising, as it could drive the natural emergence of exploration, learning, and teaching behaviors through environmental demands. We refer to discussion Sec.5.2.1 for more information on compositional dynamics as an interesting element of open-ended environments.

Notably, throughout these studies, we employed recurrent neural networks as controllers, enabling lifetime adaptation that can be meta-learned through outer adaptation loops [182–185]. This architecture theoretically supports the emergence of learning and exploration during an agent's lifetime. Such adaptation can be particularly advantageous in highly variable environments, depending on its temporal scale and intensity [186]. The next chapter will explore how this environmental variability can lead to the emergence of collective exploration and how communication can enhance exploration efficiency.

Interaction between different adaptation scales: learning to learn and to explore

3

In the last chapter, we explored how reciprocal causation between agents' behavior and environmental dynamics could result in eco-evolutionary dynamics (e.g. Lokta-Volterra cycles, Sec.2.1.4) or the acquisition of collective eco-engineering strategies (e.g. agriculture, Sec.2.2). We have seen that such phenomena could result in important variability in the environment, e.g. in terms of resource availability and distribution. In the natural world, even "stabilization" strategies, like agriculture, often come with new techniques or tools that appear at a fast pace compared to evolution. These rapid changes require equally fast adaptation mechanisms, enabling agents to adapt in shorter timescales than the ones at which biological evolution operates.

The evolution of learning, for instance, is hypothesized to have been favored in environments that are unpredictable across generations but sufficiently stable within an individual's lifetime [160, 326]. This mechanism exemplifies how an outer slow adaptation loop, evolution, can give rise to a faster adaptation mechanism: developmental learning, as a response to environmental variability. In fact, in these variable environments, despite costing time and being potentially risky, developmental learning seems to have evolved as a way to efficiently adapt to diverse possible environments whose properties cannot be predicted at the evolutionary timescale.

In particular, a fundamental aspect of learning is exploration. In humans, this process is primarily driven by intrinsic motivation (Sec.0.2.3): engaging in activities for their own sake rather than for external rewards. While intrinsic motivation has been extensively studied in psychology [327–331], its precise mechanisms and evolutionary origins remain topics of active research. Computational models of intrinsic motivation have therefore been used both for their usefulness for enhancing the exploration capabilities of artificial agents as well as to help understand the mechanism and origins of intrinsic motivation.

In silico, intrinsic motivation has been introduced in two different manners.

- The first one involves directly implementing it into an artificial agent's cognitive architecture to enhance exploratory behavior. For example, several works propose to add intrinsic bonuses to the reward function for novel state [136, 137], model prediction error [138–140] ,surprise [141], (in)competence [142], or empowerement (how much the agent can "change its environment") [143]. Another approach consists in enabling agents to generate their own intrinsic goals and learn how to achieve them, what is called autotelic agents [144, 145] (see Fig.3.1 and sec.3.2.6 for a formalization of autotelic agents). We refer to [134] for a general review of computational models of intrinsic motivations.

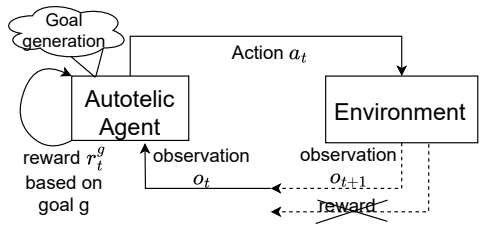


Figure 3.1: **Autotelic agent**: agents that generate their own goal and actively learn how to achieve them.

- On the other hand, several work have studied how complex exploration and intrinsic motivation can emerge from the training itself [171, 332–334]. In fact, similar to how evolutionary processes (an "outer" adaptation loop operating at the timescale of generations) gave rise to developmental learning (an "inner" adaptation loop operating at the timescale of an individual's life) in biological systems, computational studies suggest that learning itself – and in particular exploratory behavior and intrinsic motivations – can arise from an outer adaptive process. This concept is known as meta-learning (learning to learn, Fig.3.2). Meta-learning was shown to be a powerful technique, even able to meta-learn entire learning algorithms [169, 172, 173, 335]. In reinforcement learning (RL), meta reinforcement learning (meta-RL) approaches have shown efficacy in variable environments, allowing agents to generalize and adapt to new settings dynamically through behavioral plasticity and exploration [170]. In particular, several studies have demonstrated the meta-learning of powerful exploration strategies [182–185, 333, 334]; as well as the meta-learning of explicit intrinsic motivations [171, 332]. Many of these studies underscore the critical role of environmental variability in promoting the emergence of adaptive strategies, highlighting its influence on the development of exploration [184, 186].

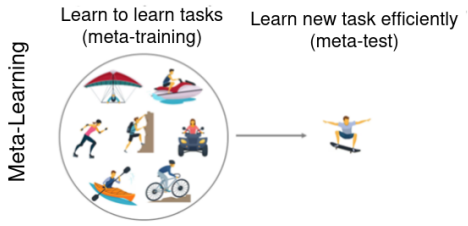


Figure 3.2: Meta-learning. Fig from [169].

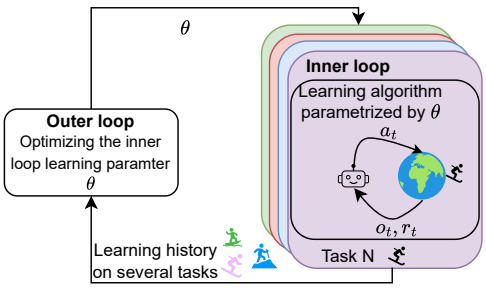


Figure 3.3: Meta-RL. An outer loop optimizes the parameters of a learning algorithm (inner loop). Fig inspired from [170].

Towards Multi-Agent Systems However, most of these works focused on single agents environments without interaction with other learning agents, or used centralized training using copies of the same agent [184]. This contrasts sharply with natural environments as we considered in the previous chapter which often assume groups of independent agents interacting. In fact, interaction between multiple learning agents can be beneficial to exploration, through the sharing of diverse experiences allowing to better escape potential local optima [154]. However, multi-agents systems also face the challenge of the variability due to other agents' change of behavior, to which the agent has to adapt [336, 337]. In particular, in environments necessitating cooperation, groups of agents have to potentially align their intentions to explore more efficiently.

In this chapter, we will study how advanced, generic, potentially collective exploration strategies can emerge in adaptive agents exposed to high environmental variability.

Emergence of Collective Exploration In a first section, using procedurally generated hierarchical tasks, we explore the emergence of **collective** exploration in a group of independent agents. From the training on a diverse distribution of tasks where the underlying rules have to be discovered, agents meta-learn to collectively explore the affordances of the environment. The agents also show interesting generalization to new tasks and longer chains of tasks (with more objects etc) not seen during training.

Communication and emergent shared intentionality In a second contribution, we now assume **group of independent agents with a built-in intrinsic motivation mechanism**. We rely on the autotelic learning paradigm (formalized in Sec. 3.2.4), where each agent is able to self-generate its own goals and learn how to achieve them using goal-conditioned RL. We first show that agents independently selecting their goals achieve sub-optimal behavior in cooperative environments. We then show that the alignment of goals is a sufficient condition to efficiently learn optimal cooperative behaviors. Lastly, we provide a fully decentralized training algorithm, which allows agents to learn to develop "shared intentionality" [338], establishing a common lexicon that enables coordinated goal selection. Notably, the learning of "shared intentionality" is done through the agents' individual maximization of reward. This work shows how communication can reduce uncertainty arising from other agents', by aligning intentions to promote more efficient exploration.

3.1 Emergence of Collective Open-Ended Exploration from Decentralized Meta-Reinforcement Learning

Context

This work is the result of the master internship of Richard Bornemann which I co-supervised.

- Bornemann*, R., Hamon*, G., Nisioti, E., Moulin-Frier, C. (2023) *Emergence of collective open-ended exploration from Decentralized Meta-Reinforcement learning*. **In Second Agent Learning in Open-Endedness (ALOE) Workshop at Neurips 2023**

I am co-first author of the paper

It was presented at the Agent Learning in Open-Endedness (ALOE) Workshop at Neurips 2023.

Abstract

Recent works have proven that intricate cooperative behaviors can emerge in agents trained using meta reinforcement learning on open-ended task distributions using self-play. While the results are impressive, we argue that self-play and other centralized training techniques do not accurately reflect how general collective exploration strategies emerge in the natural world: through decentralized training and over an open-ended distribution of tasks. In this work, we therefore investigate the emergence of collective exploration strategies, where several agents meta-learn independent recurrent policies on an open-ended distribution of tasks. To this end, we introduce a novel environment with an open-ended procedurally generated task space which dynamically combines multiple subtasks sampled from five diverse task types to form a vast distribution of task trees. We show that decentralized agents trained in our environment exhibit strong generalization abilities when confronted with novel objects at test time. Additionally, despite never being forced to cooperate during training, the agents learn collective exploration strategies which allow them to solve novel tasks never encountered during training. We further find that the agents' learned collective exploration strategies extend to an open-ended task setting, allowing them to solve task trees of twice the depth compared to the ones seen during training.

3.1.1 Introduction

Cooperative exploration plays a pivotal role in fostering the collective intelligence in groups of autonomous agents. Developing strategies to effectively coordinate the exploration of large search spaces has the potential to significantly decrease the time needed to find optimal solutions. The power of cooperative exploration can be seen in areas ranging from complex search and rescue missions to the entire field of modern science, where scientists work together to coordinate their research. Studying the emergence of cooperative behavior in artificial agents has therefore garnered much interest, especially in the field of multi-agent reinforcement learning [339] [340] [341]. With the recent successes of deep reinforcement learning, the difficulty of the tasks being researched has significantly increased, leading to a corresponding increase in the complexity of learned cooperative behaviors [114] [115]. Extending multi-agent reinforcement learning further by training agents on open-ended task spaces has led to agents which exhibit strong generalization abilities, being able to adapt to novel tasks through strong exploration priors acquired during training [44] [184].

However, these works do not study the simultaneous training of decentralized agents, but rather make use of techniques such as self play or playing against static checkpoints of other agents. We argue that this approach does not accurately reflect how autonomous agents learn together in the real world. Rather, when confronted with novel tasks in a group setting, all autonomous agents in the group are exploring and actively updating their prior beliefs, forcing them to either explicitly or implicitly coordinate their learning progress to converge to some shared strategy which allows them to effectively solve the task. Recent works such as [300, 342, 343] have shown that this coordination process can emerge in decentralized multi-agent reinforcement learning, leading to independent agents learning to solve complex tasks together.

In this work we want to further investigate the emergence of cooperative exploration strategies of decentralized agents by training them on an open-ended distribution of tasks. To this end we introduce a novel environment which is conceptually simple yet allows for a complex open-ended procedurally generated task space by dynamically combining multiple subtasks sampled from five task types to form a task tree which needs to be solved sequentially (Fig. 3.4), akin to the notion of recipes in [184]. We train two agents parametarized by independent recurrent neural networks and optimized using standard proximal policy optimization. As no information is given to the agents about which subtasks have been sampled or how and in which order they should be solved, the agents have to develop general strategies for exploring the environment, effectively learning how to learn from the information obtained by interacting with the environment throughout the episode, in order to solve novel tasks. We show that training independent decentralized agents on only multi-agent episodes leads to sub-optimal behavior of the agents, primarily due to the problem of credit assignment when rewards are shared between agents. We propose to include single-agent episodes during

training to force the agents to learn to solve tasks on their own without relying on any help from other agents. We find that training on a mixture of single and multi-agent episodes increases the agents' individual performance while simultaneously decreasing the individual performance differences between the agents, leading to a strong improvement in performance in multi-agent tasks.

Using this approach we find that decentralized agents trained in our environment learn a powerful collective exploration strategy, allowing them to solve over 70 percent of task trees encountered during training. Moreover, these powerful exploration capabilities lead to strong generalization performance when confronted with objects unseen during training, as well as on novel tasks which require complex coordination to be solved successfully at test time. Additionally, we show that the learned collective exploration strategies extend to the open-ended task setting, enabling the agents to effectively generalize to task trees with a depth of six, featuring an increased complexity of subtasks, despite being initially trained on task trees comprising only three subtasks.

3.1.2 Related Work

Cooperative behavior in multi-agent environments has long been a topic of great interest in reinforcement learning [339] [340] [341]. Recently, techniques from multi-agent reinforcement learning such as self-play have been used to train agents to human level performance in areas ranging from board games [224] to complex modern video games [114] [113]. Other works study the emergence of coordination and cooperation in populations of agents in complex competitive environments. [39] has shown that teams of agents competing against each other in the game hide and seek can develop sophisticated strategies such as tool use and even learn to exploit bugs in the environments implementation. These works make use of forms of centralized training, such as shared agent parameters or self-play to achieve their impressive results. [115] and [343] have shown that decentralized methods, when combined with population-based training, can lead to the emergence of complex shared cooperation and coordination strategies within teams of agents. [342] have further shown that a simplified approach of training independent agents without any centralized information sharing or population-based training can lead to competitive performance on the Starcraft Multi Agent Challenge [344].

In order for agents to deal with environments where the task at hand is unknown to the agent and sampled from large distribution of possible tasks, meta-learning has been proposed. Meta-learning allows an agent to learn to use its existing knowledge to quickly adapt to new tasks at test time [345]. Combining this approach with reinforcement learning and recurrent neural networks has lead to agents that are able meta-learn their own reinforcement learning algorithm, allowing them to adapt and solve novel tasks [182] [183]. Recent works have shown such Meta Reinforcement Learning (Meta RL) algorithms to be very effective resulting in multi task robots that are able to adapt to

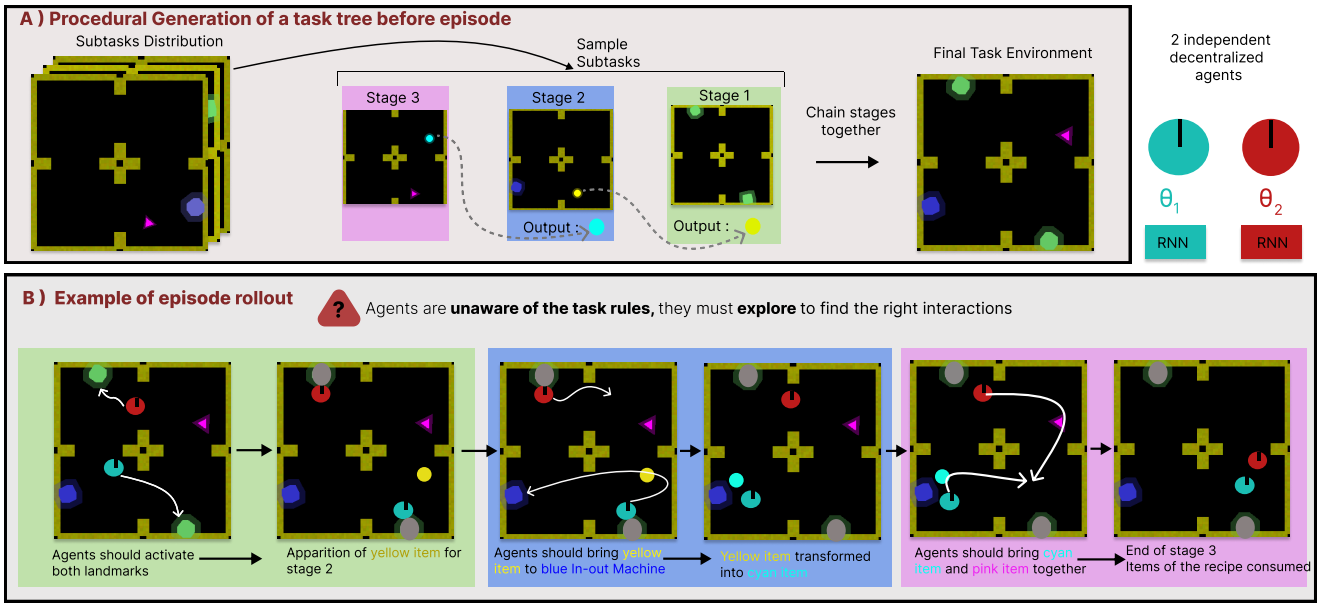


Figure 3.4: **Task Tree Sampling and Episode Rollout. A)** shows the task tree sampling process. First three subtasks are sampled from the distribution of subtasks (Section 3.1.3), one for each stage of the task tree. All of the objects required to solve the subtask for stage one and some of the objects required by subtasks in later stages are then placed in the environment. The remaining objects required to solve the later subtasks can be created through solving preceding subtasks (Section 3.1.3). **B)** shows an example of a single episode rollout. The agents have to complete the subtasks sequentially in order to create objects which are needed by the subtasks in later stages. Since a new task tree with different subtasks is sampled at the beginning of each episode and no information about the subtasks is given to the agents, the agents have to explore the environment and interact with all present objects so solve the subtask at each stage. Videos of the agents behaviors can be found on our companion website.

new tasks [346] [347] and environments [117] on the fly, even allowing them to generalize their behaviors from simulations to the real world. [196] shows that combining Meta RL with open ended procedurally generated environments facilitates open ended skill acquisition and allows agents to better adapt to novel environments. Similarly [44] and [184] show that agents trained on vast diverse task spaces are able to quickly adapt to novel tasks, even surpassing human adaptation skills.

Work in combining multi-agent environments with Meta RL has so far remained relatively sparse. [348] are exploring the use of multiple agents acting simultaneously to efficiently explore the environment in order to then leverage their pooled knowledge in the exploitation phase to solve complex tasks. [349] uses Meta RL together with open ended environment design to train a pool of agents on competitive two player tasks. Closer to our work [44] and [184] also present results for agents trained in multi agent settings in a multi task and Meta RL fashion with open ended task distributions. However, these methods employ either static checkpoints from a population of agents or older versions of themselves to train an agent in multi agent episodes. Our work therefore differs from this approach by training two decentralized agents together in the same environment, without making use of techniques like self-play or population based training, commonly used in other works on emergent cooperation.

3.1.3 Method

Environment

Our 2D environment is implemented using Simple-Playgrounds [350] and features realistic physics for object movements and collisions, as well as a range of interaction dynamics for different object types. The agents have two continuous movement actions for turning angles and forward walking speeds, as well as the two discrete grasping and activating actions for interacting with the objects present in the environment. The agents are able to pass through each other and cannot grasp an object which is currently being held by the other agent. This is done in order to limit noise caused by the agents interfering with each other during training. As input the agents get a limited top down view of their surroundings, preventing them from having full vision of the environment. The environment itself consists of rooms connected by large doorways, preventing the agents from diagonally crossing the map (Fig. 3.4). We differentiate between two object types required by the subtasks. Environment objects such as landmarks are large square shaped immovable objects that are spawned at the edges of the environment. Task objects are smaller and can be moved by the agents. Objects always have some form of interaction dynamic either with another task object, an environment object or an agent. The different interaction dynamics of environment objects, task objects and agents are explained in detail in (Section 3.1.3). The pool of possible objects includes three different shapes and colors for a total of nine different task objects, as well as a further four different environment objects. Task objects are always randomly sampled before each episode and do not poses any fixed interaction dynamic, encouraging the agents to explore by trying to combine different task objects together to create new objects. Environment objects however always possess the same interaction dynamic. Agents should therefore meta-learn how they can interact with a specific environment object.

Task types We include five different subtask types in our environment. Although all of the tasks can be solved by a single agent alone, the agents should nonetheless learn to cooperate in order to solve the tasks as quickly and efficiently as possible.

Activate Landmarks: The agents are tasked with locating one or two landmarks, which are randomly placed at the edges of the environment, and activate them. In the two landmark case, the agents have to activate both of the landmarks within three hundred environment steps of activating the first landmark. The agents are expected to learn to split up and independently locate a landmark in order to solve the task as quickly as possible.

Lemon Hunt: The agents need to find a specific object and switch into the "lemon" object by activating it. The resulting lemon object can then be consumed by either agent. The agents should learn to interact with all the task objects present in the environment and try to activate them.

Crafting: The agents need to combine two objects in the environment to either spawn a new object or make an existing object disappear. The agents are expected to first explore the environment until finding a task object of interest and trying to combine it with other task objects. Additionally the agents should coordinate to bring task objects together and not explore object combinations which have already been tried by the other agent.

In Out Machine: The agents need to find the correct object and bring it to the in out machine, which is randomly spawned at the edge of the environment. The object is then switched into an object required by following tasks. Therefore to efficiently solve this task, the agents should try bringing all the objects present in the environment to the in out machine until they find the correct object which can be switched.

Drop Off Point: Similar to the in out machine, only now the agents need to bring the correct object to the drop off point to make it disappear of completing all of the preceding subtasks.

Procedural Generation of Task Trees At the beginning of each episode, d subtasks are selected from categories of five different task types in order to build a task tree of depth d (Fig. 3.4), similar to the concept of task recipes in [184]. Initially the end condition that must be satisfied for the last task in the task tree to be considered a success is first selected, with the possible end conditions being **object exists** or **object does not exist**. The subtasks are then sampled recursively to depth d, where the subtask at each level outputs the objects required by the subsequent task. The set of subtasks which can be sampled at each stage depends on the stage and the subtasks sampled in preceding stages. At each stage, the objects required by the subtask are sampled uniformly from a pool of nine different objects. The environment objects required by all sampled subtasks and the objects required for the first subtask are then spawned in at random points in the environment. Using this method, we can build complex task trees with arbitrary depth, procedurally generating an open-ended distribution of tasks.

Training Setup

Reward Structure After successfully completing a subtask in the task tree agents are jointly rewarded for each time step until the end of the episode, encouraging them to continue improving their performance even in subtasks with very high success rates. Additionally, the reward per time step increases exponentially for completing subtasks in higher stages, incentivizing the agents to learn to better solve subtask of higher stages rather then marginally improving their performance in the easier to solve lower stages. We observe that this reward structure significantly improves performance over rewarding the successful completion of each stage the same. The reward structure in combination with the tasks trees made up of different subtasks leads to a smooth improvement of the agent across different

stages during training, eliminating the need for any explicit form of curriculum design.

Agent Architecture We employ a similar approach to [342], where each agent in the environment is independently parameterized by a neural network which is optimized using standard proximal policy optimization [320], without sharing any parts of the network. Each agent only gets as input its own limited top down view of the environment as well as its own action and reward from the previous step as usually done in Meta-RL. No additional information about the task tree, environment objects or the other agent are given to the agents. For the agent architecture we use the same convolutional neural network used in [112], followed by a one layer fully connected network of size [256] whose output is fed through a ReLU non-linearity and concatenated with the agents action and reward from the previous step. This is followed by a one layer LSTM [18] with size [256] and finally by a policy head consisting of three layers with sizes [64, 64, 4] and a value head with sizes [64, 64, 1], where each of the hidden layers is followed by a ReLU non-linearity.

Agent Training At the beginning of each episode, we first sample whether the episode will be played in the multi-agent or single-agent paradigm. In the multi-agent case, the agents will be placed in the same environment, whereas in the single-agent case, the agents will play in two different environments without interacting with each other. We then sample one task tree in the multi-agent case or two task trees, one for each environment, in the single-agent case through the procedure described in (Section 3.1.3). Since all subtasks can be solved by a single agent during training, we do not modify the task distribution for single-agent episodes. As is common in Meta-RL, the agents do not get any information about which subtasks have been sampled and in which order they should be solved. They are then randomly placed in the environment and have a limit of 1000 environment steps to solve all the subtasks sequentially. After the limit is up, the environment is reset, we resample the multi or single agent setting and sample one or two new task trees. We train on batches of 480 complete episodes for a total of 750000 episodes. We linearly decay the learning rate to 0 from a starting value of 0.00025 over the course of training. As the agents are fully decentralized, no information is shared between them during training. We therefore update the parameters of each agent's network based only on its own experiences from each batch.

3.1.4 Results

In this section, we present the experimental results for our training paradigms, both with and without single-agent episodes. We define the performance of our agents as the rate of completed subtasks per stage. We find that including single-agent episodes improves the performance during training. Further, we show that decentralized

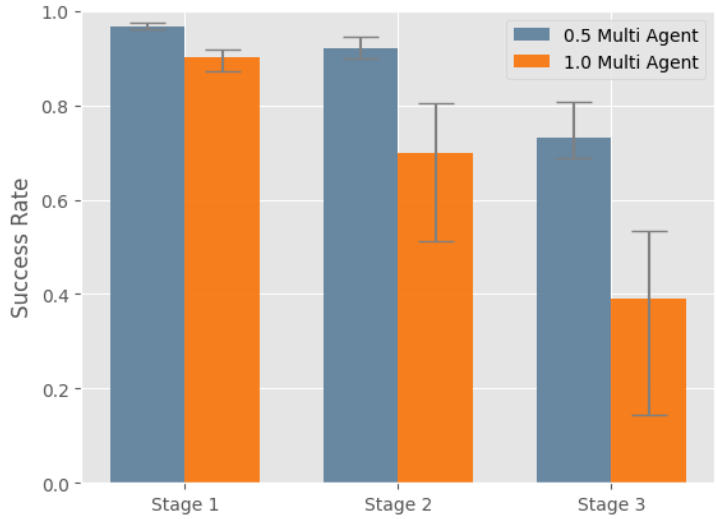


Figure 3.5: Success rates for 100% vs 50% multi agent episodes during training

agents trained in our environment exhibit strong generalization abilities when encountering objects unseen during training and complex novel tasks which require efficient coordination between the agents in order to be solved. Finally, we highlight the agents' proficiency in generalizing to the open-ended task setting. This is demonstrated through an evaluation of their performance on task trees encompassing six stages, instead of the depth of three stages encountered during training.

Training Performance

Our findings indicate that the straightforward approach of training two decentralized agents together solely on multi-agent episodes results in suboptimal performance. When looking at the individual performances of the agents during single-agent episodes, we observe significant disparities in skill levels (Fig. 3.6). We argue that these performance discrepancies stem from the credit assignment problem, which emerges due to multiple agents sharing rewards [351]. When one agent accomplishes a subtask, both agents receive the reward, resulting in potentially misleading parameter updates for the agent that was not directly involved in completing the subtask. This dynamic can magnify minor skill disparities at the outset of training, ultimately culminating in substantial differences in learned behaviors by the end of training. To combat this problem we propose training the agents on both single and multi agent episodes, as the multi-agent credit assignment problem can not arise during single agent episodes. We find that this greatly decreases the skill differences between the agents and increases the single agent performances (Fig. 3.7), leading to a large gain in performance in multi agent episodes. While agents trained on multi and single agent episodes are able to solve the vast majority of subtasks for all stages within the time limit, agents trained solely on multi agent episodes perform worse on stages one and two and fail to solve subtasks in stage three in the majority of episodes. The large dropoff in success rate from stage two to stage three is mainly caused by the agents running out of time. This indicates that

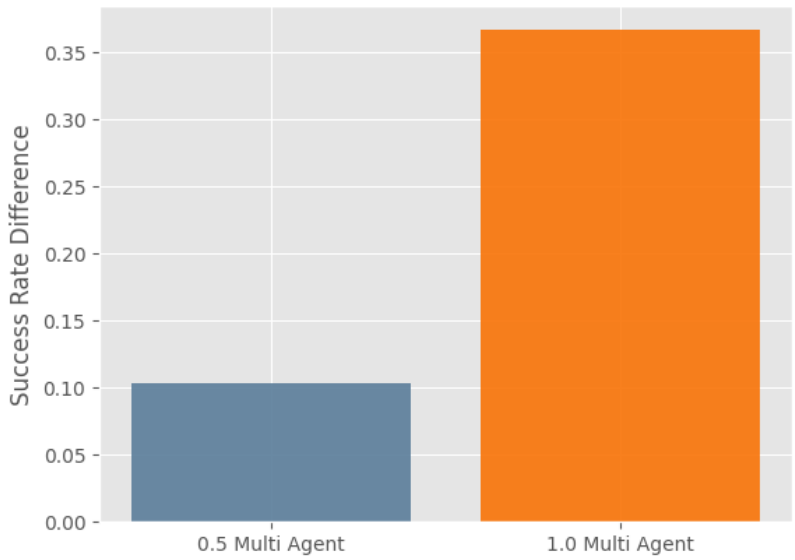


Figure 3.6: Mean stage 3 success rate difference between two agents trained on 100% vs 50% multi agent episodes

agents trained on multi and single agent episodes are able to solve the subtasks much quicker when compared with agents that where only trained on multi agent episodes. In the following we therefore limit our evaluations to the case of mixed single and multi agent training.

Generalization Performance

Novel Objects We replace all task objects present during training with novel shapes and colors and evaluate the agents performance on the training task distribution in multi-agent episodes. We find that including novel task objects does not lead to any decrease in performance when compared to the training performance with the standard task objects (Fig. 3.8). When looking at videos of the agents, we observe that they interact with novel task objects in the same fashion as they would with task objects seen during training. They explore the possible task object combinations to find environment objects or other task objects which lead to a successful interaction. As the colors and shapes of task objects do not carry any information about the task objects interaction possibilities, we suspect that the agents learn to not focus on any specific characteristics of the task objects. Instead, the agents rely on powerful exploration to try out all possible object interactions in the environment to solve tasks, allowing them to generalize seamlessly to novel objects.

Forced Cooperation To test the ability of the agents to effectively cooperate, we devise three subtasks which forcibly require the agents to cooperate in order to be solved, based on the subtasks presented in (Section 3.1.3). We call these tasks "forced cooperative". Detailed descriptions for the forced cooperation subtasks can be found in the Appendix A.5.1. During evaluation we switch out the landmarks and

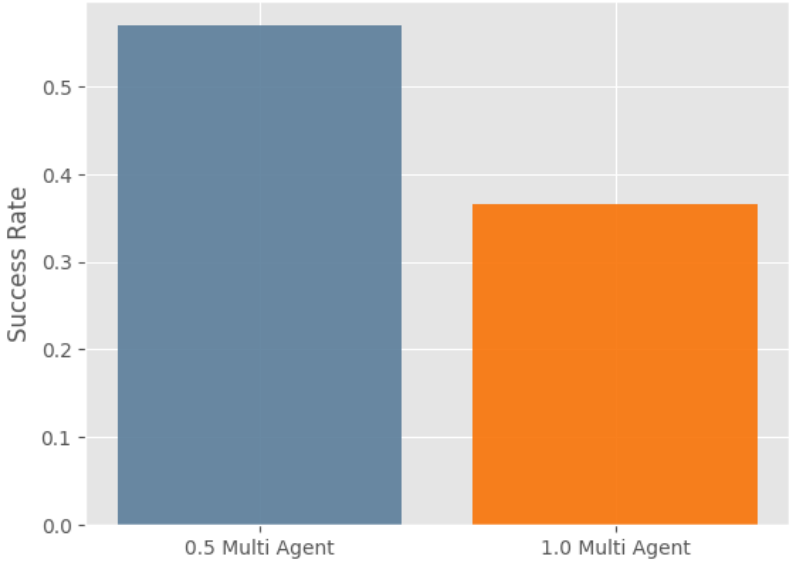


Figure 3.7: Mean stage 3 success rate on single agent episodes for agents trained on 100% vs 50% multi agent episodes

lemon hunt subtasks for these three forced cooperation subtasks and present the average results over 4800 episodes. We observe that the agents still show strong performance when it is required for them to cooperate in order to solve tasks, even with the tasks now requiring much more intricate coordination for successful completion (Fig. 3.9). This suggests that the cooperative behaviors learned during training are able to help the agents generalize to settings where cooperation is required, despite never encountering such as scenario during training. Notably, the performance of agents trained without any forced cooperative tasks does not differ significantly from the performance of the agents trained only in the forced cooperative setting, when evaluated on forced cooperative tasks. We hypothesize that this is due to the high difficulty of the forced cooperation tasks preventing the agents to efficiently learn without any form of curriculum. When analyzing the agents behavior, we observe that when confronted with novel behaviors of previously seen environment objects like landmarks, they try to exploit the behaviors learned during training. However, after some amount of unsuccessful tries, the agents start exploring the environment until finding an environment or task object which they are able to interact with. Paired with the agents behavior to often first explore the environment separately, they are able to solve complex coordination tasks like the forced cooperation version of the landmarks subtask, where the agents have to find and activate their respective version of the landmark within ten environment steps of each other. To gain a better understanding of how the agents are able to solve forced cooperation tasks where refer the reader to the accompanying website with videos

Novel Task To further evaluate the agents cooperative abilities and their capacity to generalize to novel tasks we introduce the "pressure plate" task, which the agents have never seen during training. In this task, one agent is tasked with remaining in proximity to the pressure plate landmark and continuously activating it. The second agent is responsible for locating a task object and transporting it to the "in

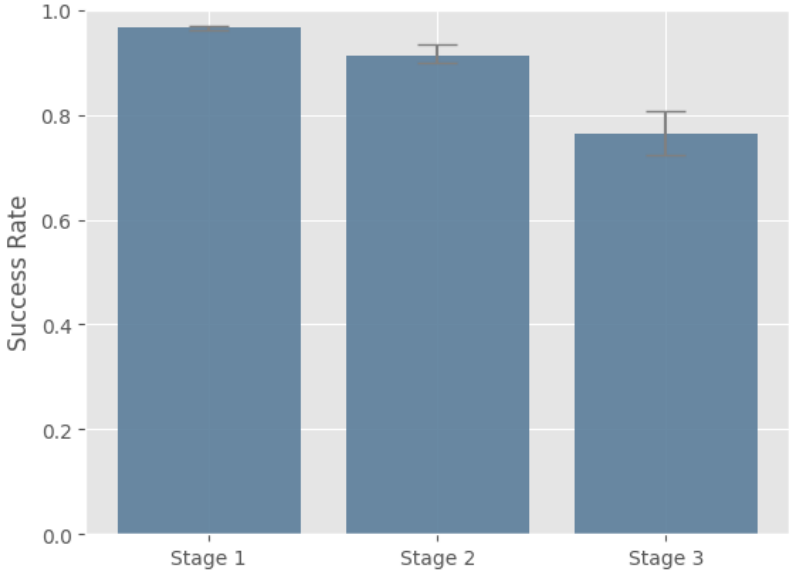


Figure 3.8: Success rates with novel objects

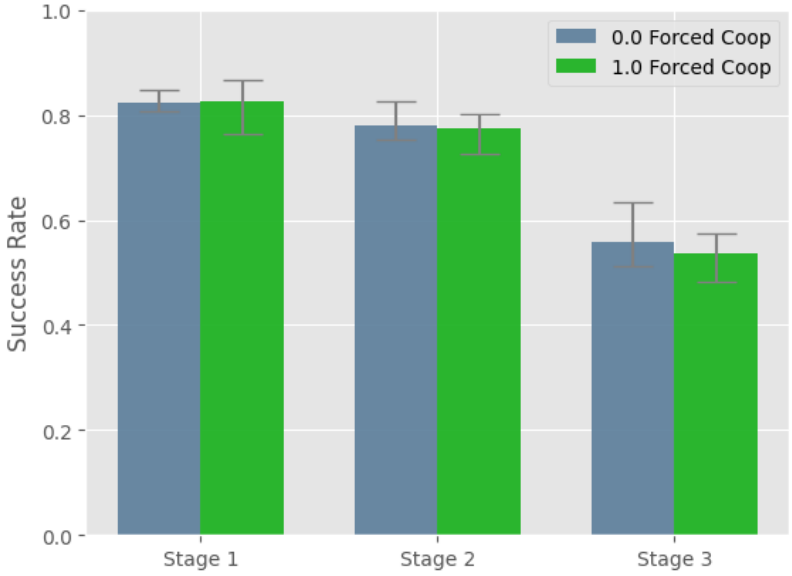


Figure 3.9: Evaluation for agents trained on 0% vs 100% forced cooperation tasks

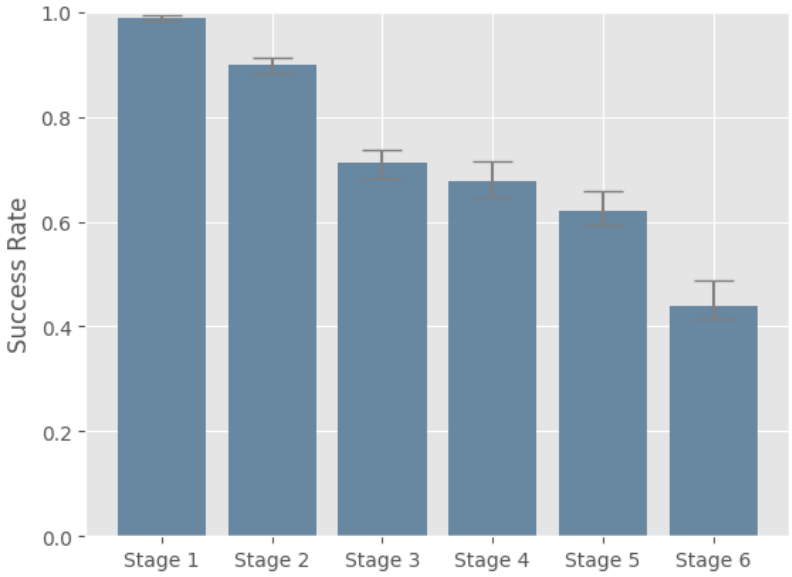


Figure 3.10: Success rates for open ended exploration on 6 stages and 4000 timesteps

and out" machine, which only works if the pressure plate is activated, in order to switch it to the condition object which indicates successful completion of the task. This task requires coordination between the agents to determine which one remains at the pressure plate and which one explores the environment. We find that the agents are able to solve this task on 45 percent of the trials, showing impressive generalization abilities. Observing videos of the agents playing this task, we notice a recurring pattern wherein the agents activate the pressure plate and promptly move away. Once the pressure plate deactivates, the agents quickly return to reactivate it, resulting in a repetitive loop where the agents continuously trigger the pressure plate. While both agents frequently become stuck in this loop initially, we observe in many instances that one agent manages to break free and begins to explore the environment. Ultimately, this agent successfully transports the task object to the "in and out" machine, thereby effectively solving the pressure plate task. We hypothesise that the behavior for one agent to break the loop arises during training when agents learn that they should split up their efforts in order to efficiently explore the environment and solve tasks as quickly as possible, similar to the behaviors observed in (Section 3.1.4). The agents ability to generalize to novel forced cooperation tasks therefore seems to mainly stem from their exploration and coordination abilities.

Open Ended Exploration We evaluate the agents ability to open endedly explore their environment by setting the number of stages in the task tree to six and increasing the time limit to 4000 environment steps (Fig. 3.10). We further set the rewards for completing each subtask to zero in order to prevent giving the agents a reward feedback for stages higher than three, which they have not seen during training. We observe that this boosts performance on higher stages during evaluation. We find that agents generalize surprisingly well to task trees with six stages. It is worth emphasizing that as we increase the number of stages in the task tree, the number of task objects present in the environment also increases. Consequently, this exponentially expands the number of possible combinations of task objects that the agents must experiment with in order to solve the subtasks. The agents strong performance therefore not only shows their capacity for collective open ended exploration but also showcases their capacity to tackle subtasks of higher complexity compared to those encountered during training.

3.1.5 Conclusion

In this work we investigate the emergence of collective exploration strategies in decentralized agents trained on an open ended distribution of tasks in a Meta RL fashion. While previous related works have studied how cooperative behaviors can emerge from Meta-RL in an open-ended task space [44][184], our approach is, to our knowledge, the first attempt at demonstrating it in a decentralized training paradigm, together with available open source code for reproducibility. We show that decentralized agents trained only on multi

agent episodes exhibit subpar performance and propose to incorporate single agent episodes to boost the individual agents performance. Agents trained using this approach exhibit strong generalization abilities to unseen objects and tasks requiring the agents to cooperate, indicating the emergence of collective exploration strategies despite never being forced to cooperate during training. We further show that the agents are able to generalize their exploration behavior to an open ended setting and solve task trees of twice the length compared to task trees seen during training. We observe that withholding any reward feedback from the agents at test time boosts the exploration performance, suggesting the emergence of an intrinsic motivation to open endedly explore the environment, even in the absence of extrinsic rewards. However, directly isolating which learned behaviors allow the agents to cooperate and coordinate their movements remains difficult. Adding a direct communication channel between the agents could therefore present a promising method to boost the agents multi agent performance and allow for a clearer understanding of their learned cooperative behaviors by analyzing the learned communication. Additionally, incorporating more sophisticated approaches to solving the multi agent credit assignment problem and investigating how to boost the agents individual performance while preserving their ability to cooperate holds potential to greatly increase the complexity of the agents learned cooperative behaviors.

Finally, looking at the learned behaviors (see videos), the agents seem to change the target interaction that they try periodically. This suggests the emergence of a proto-goal selection mechanism within the agent's recurrent policy, potentially in the space of object or interaction between objects. Further work needs to be done to confirm (or infirm) this emergence of implicit internal goal selection.

In the next chapter, we will consider agents pre-equipped with a goal generation mechanism, i.e. autotelic agents. We will experimentally show the crucial role of goal alignment in efficiently learning cooperative tasks in this setting and will explore how communication can help agents to coordinate their exploration.

3.2 Autotelic Reinforcement Learning in Multi-Agent Environments

Context

This work began with the master internship of Elías Masquil which I co-supervised. After the end of Elías's internship we pursued the experiments and writing with Eleni Nisioti. In particular, I conducted the majority of the experiments and training reported in this contribution.

- Nisioti*, E., Masquil*, E., Hamon*, G., Moulin-Frier, C. (2023). *Autotelic Reinforcement Learning in Multi-Agent Environments*. **In Conference on Lifelong Learning Agents** (pp. 137-161). PMLR.

I am co-first author of the paper

I presented this work at the Conference on Lifelong Learning Agents (Collas) 2023 in Montreal.

Abstract

In the intrinsically motivated skills acquisition problem, the agent is set in an environment without any pre-defined goals and needs to acquire an open-ended repertoire of skills. To do so the agent needs to be *autotelic* (deriving from the Greek *auto* (self) and *telos* (end goal)): it needs to generate goals and learn to achieve them following its own intrinsic motivation rather than external supervision. Autotelic agents have so far been considered in isolation. But many applications of open-ended learning entail groups of agents. Multi-agent environments pose an additional challenge for autotelic agents: to discover and master goals that require cooperation agents must pursue them simultaneously, but they have low chances of doing so if they sample them independently. In this work, we propose a new learning paradigm for modeling such settings, the Decentralized Intrinsically Motivated Skills Acquisition Problem (Dec-IMSAP), and employ it to solve cooperative navigation tasks. First, we show that agents setting their goals independently fail to master the full diversity of goals. Then, we show that a sufficient condition for achieving this is to ensure that a group *aligns* its goals, i.e., the agents pursue the same cooperative goal. Our empirical analysis shows that alignment enables specialization, an efficient strategy for cooperation. Finally, we introduce the Goal-coordination game, a fully-decentralized emergent communication algorithm, where goal alignment emerges from the maximization of individual rewards in multi-goal cooperative environments and show that it is able to reach equal performance to a centralized training baseline that guarantees aligned goals. To our knowledge, this is the first contribution addressing the problem of intrinsically motivated multi-agent goal exploration in a decentralized training paradigm.

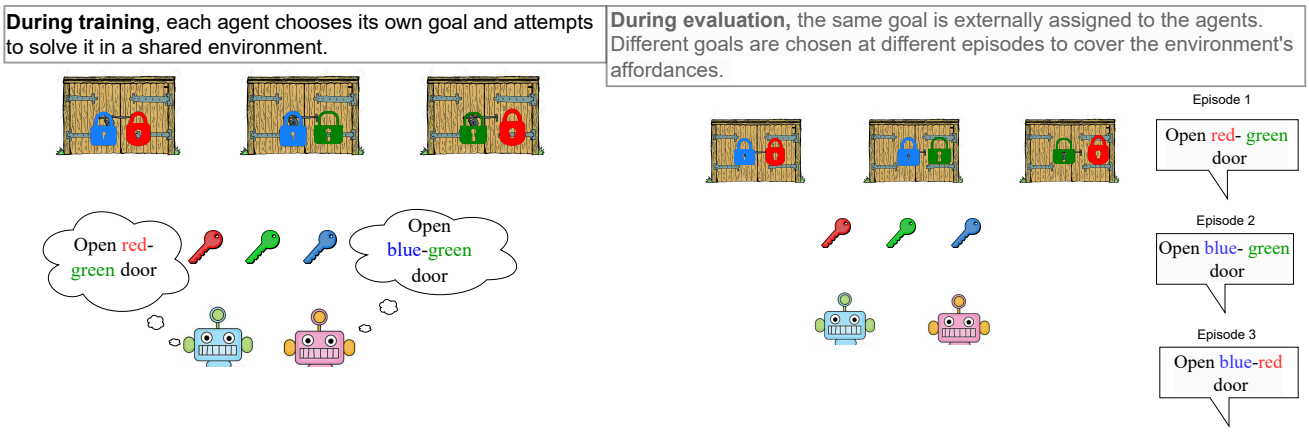


Figure 3.11: Illustrative example of learning in a Dec-IMSAP: two agents are in a shared environment where goals have the form of doors that open upon matching each lock with the key of the same color. An agent can carry at most one key, so it takes two to open a door. (Left) As agents are sampling their own goals, the group may experience episodes that cannot be solved by at least one agent: if the blue agent picks the red key and the pink agent picks the green then the blue agent will succeed and the pink will fail. (Right) During evaluation agents are assigned with the same goal.

3.2.1 Introduction

Many multi-agent scenarios require the cooperation of agents with a rich diversity of skills. Multi-player games such as StarCraft [352] and Capture the Flag [115] and real-world scenarios such as cooperative navigation in teams of robots [353], require agents that can coordinate their actions in the face of continuously-arising new challenges. When alone, a reinforcement learning (RL) agent can acquire a wide diversity of skills by being *goal-conditioned* [123] and *intrinsically motivated* [135, 144, 354]: the former means that the agent can pursue different goals at different times and conditions its learning on its current goal, while the latter means that these goals are generated by the agent using some internal reward mechanism instead of being externally set by the human designer. Such agents have been termed autotelic [144]. But what happens when you place multiple autotelic agents in the same room, expecting them to autonomously discover all the room's affordances? We argue that you will stumble upon a challenge: for agents independently generating their own goals, the probability of sampling the same one reduces dramatically with the size of the goal space. Thus, we expect that these agents will fail to master goals that require cooperation (such as lifting a heavy box), as they will collectively pursue them rarely and receive a noisy training signal due to the fact that the goals of others are not directly observable. In this work, we introduce a new type of problem for multi-agent RL, the Decentralized Intrinsically Motivated Skills Acquisition Problem (Dec-IMSAP), to capture such settings, propose a decentralized algorithm for tackling it, the Goal-coordination game, and evaluate it in a cooperative navigation task [1].

For single-agent settings, autonomous skill discovery has been formalized as the Intrinsically Motivated Skills Acquisition Problem [144]. To meaningfully extend it to multi-agent settings we need to consider environments that require cooperation: some of the goals will be cooperative, i.e., at least one other agent needs to act for the agent to achieve its goal, while the rest will be independent, i.e., they

1: We provide code for reproducing the experiments presented in this contribution at `https://github.com/Reytuag/imgc-marl`

can be achieved independently of others. To study the Dec-IMSAP we propose a new training/evaluation paradigm: during the training phase agents are autonomously setting their own goals and learning to achieve them in a fully-decentralized manner, while, during evaluation, we externally provide agents with the same cooperative or individual goal, ensuring that a wide diversity of goals is tested across evaluation episodes. We provide an illustrative example of this problem setting in Figure 3.11.

Intrinsic motivation originated in the field of cognitive science, with studies focusing on human infants due to their impressive ability to efficiently learn new skills [135]. Explanations rely on exploratory play, during which infants generate their own goals and learn how to achieve them for the mere purpose of discovering new learning situations [329, 355]. While studies primarily consider a single human subject, some study infants engaging in cooperative play and show that they can plan alongside others [356–358]. According to theories of human social intelligence [338], we may owe our ability to cooperate more extensively than other species to our *shared intentionality*: to solve tasks that require cooperation we attend to the same goal with others and know that we are doing so.

Does shared intentionality play an equally important role in groups of artificial agents and, if so, how can we guarantee it in a fully-decentralized training regime? This is the main research question we aim to address with our study of the Dec-IMSAP. As we more concretely explain in Section 3.2.6, we are motivated by real-world applications, such as groups of cleaning robots or disaster robotics, where a group needs to adapt to a diversity of tasks, some of which may require cooperation. To address this question empirically, we study a simplified two-player setting with goal-conditioned RL agents [144] that sample goals randomly from a fixed, pre-defined set. Such agents have been previously extended to multi-agent settings assuming external supervision during training [359], thus not considering autonomous learning. We measure the degree of shared intentionality as *goal alignment*, a metric quantifying the percentage of training episodes during which two agents pursue the same cooperative goal. First, we artificially control for the level of alignment and observe it is highly correlated with performance. Then, we propose the Goal-coordination game, a fully-decentralized emergent communication algorithm that enables goal coordination during training. Under the Goal-coordination game, before acting, an agent, chosen at random, takes the role of a leader, samples its own goal and communicates a message to the follower. which selects its own goal based on it. Crucially, the agents learn how to map goals to messages and vice versa by purely maximizing their individual rewards. By coordinating in the message, rather than the goal space, agents using the Goal-coordination game can align their goals even if they employ different goal representations. We show that alignment emerges so that the population reaches equal performance to a centralized setting that guarantees alignment. To get a clearer understanding of the temporal dynamics of the Goal-coordination game, we analyze the co-evolution of messages, goal alignment and group rewards and discover that interesting collective behaviors emerge. Our contributions are:

1. the formulation of the Dec-IMSAP, a new type of problem for studying intrinsic motivation in multi-agent systems with goal-conditioned RL agents;
2. a detailed analysis on the impact of goal alignment between agents in the Dec-IMSAP;
3. an algorithm for solving the Dec-IMSAP, the Goal-coordination game, that enables agents in a group to acquire a large repertoire of cooperative skills in a fully-decentralized setting by learning how to communicate about their respective goals.

We discuss related works in Section 3.2.2 and, in Section 3.2.3, provide definitions from existing works in single-agent and multi-agent RL that we built upon to formulate the Dec-IMSAP. We, then, present our formal definition of the Dec-IMSAP and the algorithm that we propose for solving it, the Goal-coordination game in Section 3.2.6. In Section 3.2.7, we empirically analyze the behavior of groups of agents, where we employ cooperative navigation tasks as instances of the Dec-IMSAP. Finally, we discuss limitations and future directions for our work in Section 3.2.8.

3.2.2 Related Works

Early in RL development, real-world applications, such as human-interfacing robots, pushed for algorithms that can solve, not just a static task, as classically assumed by the framework of Markov Decision Processes, but tasks that change with time [360]. This problem setting, termed multi-goal RL, has been formulated under frameworks such as options and skills [361, 362], goal-conditioned policies [144, 363] and universal value function approximators [364]. Multi-goal RL has been extended to multi-agent settings, such as cooperative navigation in fleets of robots, under the framework of multi-goal Markov games [359, 365]: a group of agents employing goal-conditioned policies is trained under the supervision of a human designer that selects which goal each agent needs to pursue during each training episode.

Real-world applications soon posed another, more stringent requirement on RL: tasks do not just vary with time but may change in unpredictable ways. Thus, autonomously discovering new tasks and adapting online to them became part of the problem [366, 367]. Inspiration for tackling this setting was found in child development, in particular in exploratory play, during which infants showcase an impressive ability to self-generate their own goals and autonomously learn how to achieve them. This mechanism has been termed intrinsic motivation and has inspired a variety of unsupervised learning objectives for RL algorithms in both single-agent [140, 250, 368, 369] and multi-agent [370] settings. Not all intrinsically-motivated agents are goal-conditioned [140, 369, 370]. The ones that are, termed autotelic [144], are particularly interesting for real-world applications, as they precisely capture multi-goal settings with goals set and mastered autonomously by the agent. If we attempt to transfer autotelic RL to multi-agent settings, we see that multi-goal Markov games are not adequate. By assuming that an external supervisor is deciding which

goals will be pursued and how they will be divided among agents, it bypasses the main question in open-ended settings: how can the agents set goals autonomously? Taking into consideration that some of these goals may require the coordination of multiple agents, autotelic learning becomes more challenging when transferred to multi-agent setups.

Although previous works studied the interaction between autotelic agents and social partners [251, 371], we first propose here to study the coordination of goals set by a group of autotelic agents without any prior knowledge. For this, we can draw inspiration from the problem of intra-episode action coordination, a long-standing subject in MARL. For example, in a cooperative navigation task, others may act as obstacles that affect the observations perceived by an agent in an unpredictable way. To tackle this problem, algorithms may choose to learn a centralized critic while keeping policies decentralized [372, 373], model the behaviors of others [353] or learn to communicate [374–376]. This last approach is particularly promising when the group is heterogeneous, as coordination takes space in an abstract, learned space. Recent emergent communication algorithms have focused on action selection during the episode, while our work considers communication for goal selection before the episode starts. This is a novel problem that we study in a simplified setting and can benefit from earlier algorithms in the field developed precisely for studying the emergence of shared lexicons in grounded settings [376].

3.2.3 Background

We first describe the problem of autonomous skill acquisition in single-agent settings as an evolution from classical RL to goal-conditioned and intrinsically-motivated agents in Section 3.2.4. Then, in Section 3.2.5, we discuss multi-goal Markov games as a generalization of MARL to goal-conditioned settings with externally-provided goals .

3.2.4 Intrinsically motivated goal-conditioned reinforcement learning

In RL an agent observes an environment and performs actions on it that incur rewards, aiming at maximizing the rewards it accumulates. This interaction is commonly formalized as a Markov Decision Processes (MDP): at each time step t of an episode that lasts for T time steps the agent observes the environmental state s_t, performs action a_t and receives reward r_t. The policy $\pi(a_t|s_t)$, which describes the agent's behavior as a mapping from states to actions, is interactively learned from experience by maximizing the cumulative reward $G_t = \sum_{t=0}^{T} \gamma^t r_t$, where γ is a parameter quantifying how heavily future rewards are discounted [108]. Formally, we denote an MDP as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \rho_0, R)$, where the state space $\mathcal{S}$ and action space $\mathcal{A}$ indicate all possible configurations for the state and action respectively, $\mathcal{T}(s_{t+1}|s_t, a_t)$ is the transition function that controls the distribution of the next state s_{t+1} from the current state s_t when the agent takes

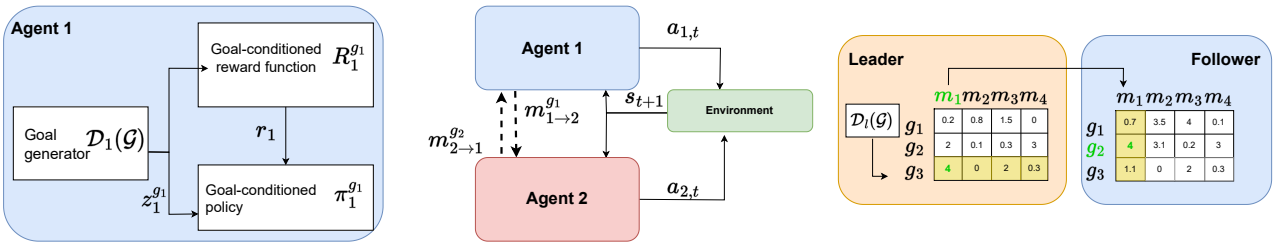


Figure 3.12: (Left) Illustration of an autotelic agent equipped with: a goal-sampling distribution $\mathcal{D}_n(\mathcal{G})$ for selecting its own goal at the beginning of each training episode, the goal-conditioned policy π_n^g and reward function $R_n^{\mathcal{G}}$. (Middle) Two autotelic agents in a shared environment, trained in a fully-decentralized manner and able to exchange messages m through a discrete communication channel that helps them coordinate their goal selection. (Right) Illustration of the Goal-coordination game: each agent maintains its own matrix associating goals to messages (see Section 3.2.6 for a description of how this matrix is learned). After the leader and follower roles are randomly assigned to the two agents, the leader samples its own goal (e.g. g_3) and transmits the message sampled from a softmax on the corresponding row (m_1). The follower samples the goal from a softmax on the corresponding column (g_2)

action a_t, ρ_0 is the distribution over the initial states, and the reward function $R(s_t, a_t)$ determines the reward that an agent receives at each time step for a given state-action combination.

Agents may need to reward themselves differently based on the task they are currently occupied with. For example, if we imagine a cleaning robot in a household, then the action "turn on the oven" should be rewarded if the robot's task is to prepare food but penalized if the task is to make sure the tenant can safely leave for a weekend trip. To expand MDPs to suit this multi-task nature of problems, the goal-conditioned RL paradigm introduces the notion of a *goal* and conditions the reward function and policy on it. Formally, a goal g is a tuple (z^g, R^g), where z^g is a goal embedding, R^g denotes the goal-conditioned reward function and π^g the goal-conditioned policy [144, 360, 364, 377]. We can, thus, define a multi-goal MDP as a set of MDPs that share $\{\mathcal{S}, \mathcal{A}, \mathcal{T}, \rho_0\}$ and differ only in the reward function R^g. We denote the space of possible goals as $\mathcal{G}$.

Multi-task learning is necessary but not sufficient for open-ended learning. In the latter, the agent needs to master not just multiple tasks but a continuously increasing set of tasks, potentially unknown at the time of the agent's design. Thus, in contrast to the classical goal-conditioned setting where goals are *externally provided*, the agent will need to generate them itself, through what is called intrinsically-motivated goal exploration[250, 378–380]. This setting has been termed as the Intrinsically-Motivated Skills-Acquisition Problem and the agents that can solve it as autotelic agents [144]. Contrary to the classical RL paradigm where a reward function is part of the environment, an autotelic agent encapsulates the reward function R^g, alongside with a mechanism for sampling goals from the goal space, the goal-sampling function $\mathcal{D}_{\mathcal{G}}$. We provide an illustration of an autotelic agent on the left of Figure 3.12.

3.2.5 Goal-conditioned multi-agent reinforcement learning

In multi-agent RL N agents interact in a shared environment, a setting that can be formalized as a Markov Game [381]. The group's behavior is captured by the joint action $\vec{a}_t = \langle a_{1,t}, \cdots, a_{N,t} \rangle$ where n indicates agent's index. After all actions are executed, the environment returns the next state s_{t+1} and a local reward for each agent $r_{n,t} = R_n(s_t, \vec{a}_t)$. To model decentralized learning in a Markov Game we can employ the framework of decentralized partially-observable MDPs (Dec-POMDPs) [382]. *Decentralization* characterizes multi-agent systems where agents do not have access to the observations of others and *partial observability* refers to the fact that this local information may not be sufficient to infer the environment's state, which now includes the other agents. To capture partial observability POMDPs introduce the notion of an observation O_n which maps the environmental state to a local observation for agent n. Formally, a Dec-POMDP is modeled as a tuple $(\mathcal{N}, \mathcal{S}, \{\mathcal{A}_n\}, \mathcal{T}, \{\mathcal{R}_n\}, \{\mathcal{O}_n\})$, where $\mathcal{N}$ denotes the set of agents and $\mathcal{A}_n$ and $\mathcal{O}_n$ are the action and observation space of a single agent.

Multi-goal Markov Games extend Markov Games to goal-conditioned settings [359]. They arise when we replace the reward function with one conditioned on goals that is shared by all agents: $r_{n,t} = R(s_t, \vec{a}_t, g_n)$. In multi-goal Markov Games goals are externally provided by a supervisor. Each agent has one fixed goal, only known to itself, and rewards are individual even though the reward function is shared, as they are conditioned on goals.

3.2.6 Autotelic agents in goal-conditioned games

Motivation

How can a learning framework model a group of agents whose objective is to learn a diversity of goals in a shared environment without external supervision? Autotelic learning well captures autonomous skill acquisition but does not consider interactions between multiple agents. Multi-goal Markov Games, on the other hand, model interactions of co-existing goal-conditioned agents but do not account for the fact that agents may be generating their own goals. We refer to this problem setting as the *Decentralized Intrinsically Motivated Skills Acquisition Problem* (Dec-IMSAP) and, in the following, provide a formal definition for it.

Formalization

A Dec-IMSAP is modeled as a tuple $(\mathcal{N}, \mathcal{S}, \{\mathcal{O}_n\}, \{\mathcal{A}_n\}, \mathcal{T}, \{R_n^g\}, \{\mathcal{D}_n(\mathcal{G})\})$, where $\mathcal{N}$ is the set of N agents, $\mathcal{S}$ is the state space, denoting all the possible configurations of all N agents and the environment, O^n and $\mathcal{A}^n$ are the observation and action space for a single agent, $\mathcal{T}(s'|s, \vec{a})$ the transition function, R_g^n is the goal-conditioned reward function

and $\mathscr{D}_n(\mathscr{G})$ is the goal-sampling distribution of agent n. Note that, differently from multi-goal Markov Games, described in Section 3.2.5, the reward function is not shared among agents. This is because, as we described in 3.2.4, in intrinsically-motivated learning the reward function is internal to the agent and, thus, may differ across agents. Also, we assume that the goal space $\mathscr{G}$ which contains all possible goals g, is known and identical for all agents. We illustrate two autotelic agents in a shared environment in the middle of Figure 3.12.

At the beginning of a training episode, each agent n samples its own goal $g_n \in \mathscr{D}_n(\mathscr{G})$, executes its goal-conditioned policy $\pi_n^{g_n}$ and adjusts its behavior to maximize the cumulative reward using goal-conditioned RL. After a fixed number of training iterations, agents will be evaluated over all possible tasks in a coordinated fashion. By "coordinated" we mean that, during evaluation, agents are assigned with the same, randomly-sampled goal. By doing so, we ensure that there is a fair evaluation of the group's ability to solve all possible cooperative tasks. Agents will be evaluated on the cumulative reward they get and the time they take to solve the goal.

The Dec-IMSAP is a problem formulation that can well capture the need for autonomous skill acquisition in teams of robots employed in real-world applications. For example, picture a group of assistance robots employed by a company to clean their offices. Naturally, the team is expected to execute various tasks, some of which may require a single robot while others may require multiple of them (for example carrying a heavy table to another room). How can the company be certain that the robots can execute any possible task when asked to? Following an externally-supervised training paradigm, the company could list all anticipated tasks, assigning a sub-task to each agent [359]. But this approach quickly becomes impractical once one acknowledges that the list may be large and change in unanticipated ways. Under the Dec-IMSAP, we propose an unsupervised training paradigm to exactly tackle these challenges. In this example, the group of agents is left for some time in the offices to discover all their affordances and learn how to solve them. In our proposed solution, the robots can come from different manufacturers, as they learn a communication protocol that allows them to coordinate even if their goal representations differ.

To solve the Dec-IMSAP the agents must learn how to solve a wide diversity of cooperative goals during training. Since both goal selection and training are decentralized this is not guaranteed: if agents sample their goals independently then some cooperative goals may not be pursued enough times during training for the group to learn how to achieve them. In addition, the reward feedback is noisy: even if agents have learned optimal policies for all cooperative goals, they can obtain zero reward if their sampled goals are inconsistent (a case we have illustrated on the left of Figure 3.11).

The Goal-coordination game

We would like to introduce a process that allows the agents to coordinate their goals without introducing centralization nor pre-existing

Algorithm 1: Goal-coordination game

```
1  Input: Population: 𝒫 = [n_1, ···, N], matrix update rate α, message space size M, goal space
     size G, batch size B
2  for agent n ∈ 𝒩 do
3      Initialize C_n = zeros(G, M) ;                          /* Initialize matrices */
4  while not converged do
5      rollouts = [];                                          /* Collect a batch of episodes */
6      for episode ∈ {1, ···, B} do
7          l = sample(𝒫) ;                                     /* Randomly select leader */
8          f = sample(𝒫 − l)
9          g_l = l.chooseGoal() ;                              /* Sample goal with 𝒟_𝒢 */
10         m_l = softmax(C_l[g_l, :])
11         m_f = m_l
12         g_f = softmax(C_l[:, m_f])
13         rollouts.append(collectRollout(l, f, g_l, g_f)) ;   /* Run a single episode */
14     for agent n ∈ 𝒩 do
15         Initialize update_n = zeros(G, M);Initialize norm_n = zeros(G, M)
16     for rollout ∈ rollouts do
17         for agent n ∈ 𝒩 do
18             n.trainPolicy(rollouts) ;  /* Update policies with new experience */
19         n_1, g_1, m_1, r_1 = rollout.l, rollout.g_l, rollout.m_l ;   /* Unpack information */
20         n_2, g_2, m_2, r_2 = rollout.f, rollout.g_f, rollout.m_f
21         update_{n_1}[g_1, m_1]+ = r_1; norm_{n_1}[g_1, m_1]+ = 1;update_{n_2}[g_2, m_2]+ = r_2;norm_{n_2}[g_2, m_2]+ = 1
22     for agent n ∈ 𝒩 do
23         C_n = (1 − α) · C_n + α · update_n/norm_n ;          /* Apply matrix update */
```

knowledge within the group and is flexible enough to deal with any behavior arising during training. To achieve this, we propose an algorithm inspired from the Naming Game, an algorithm originally introduced to help a population of agents invent a shared lexicon [376]. Our proposed algorithm, whose pseudocode we present in Algorithm 1, takes place right before an episode starts. As is common in emergent communication literature, it employs two agents and can be extended by considering a population of agents and randomly sampling a pair of them at each episode. Each agent is equipped with a communication matrix $C_n : |\mathscr{G}| \times |\mathscr{M}| \rightarrow \mathscr{R}$, where $\mathscr{G}$ is the goal space and $\mathscr{M}$ is a message space, where we consider that both spaces are discrete (we discuss in Section 3.2.8 an extension to continuous spaces). Each row of matrix C_n corresponds to a different goal g of the agent and each column to a different message m. All values of the tables are initialized with zeros (line 3). Communication is asymmetric: at the beginning of the goal-coordination round one agent is randomly chosen to be the leader and the other the follower (lines 8 and 9), therefore ensuring that each agent takes both roles across episodes. In what follows we employ underscore l to denote properties of the leader and underscore f for the follower. When an agent is the leader, the entries of its matrix answer the question: "What reward do I expect in this episode if I transmit message m when I have goal g? ". When an agent is the follower, the question is: "What reward do I expect in this episode if I choose goal g when I receive message m? ". Thus, an agent maintains a single matrix that it employs both as a leader (to infer a message given a goal) and a follower (to infer a goal given a message).

The leader first samples a goal g_l according to its own goal sampling strategy, $\mathscr{D}_l(\mathscr{G})$ (line 10), and, then, transmits the message $m_{l \rightarrow f}$ chosen using a softmax over the corresponding row of C_l (line 11). The follower receives message $m_{l \rightarrow f}$ and applies a softmax on the corresponding column C_f to pick its own goal (line 13). After playing

several episodes every agent updates its matrices to reflect the average reward for that specific goal/message association computed on the batch of collected episodes collected (lines 20-29). Note that during an episode the leader and follower may be pursuing different goals, so although they experience the same episode their rewards may differ. To ensure that the matrix updates are not too quick for an agent to adapt its policy, we employ an exponential moving average update function with update rate α (line 32). We illustrate a single round on the right of Figure 3.12. We should emphasize that agents in the Goal-coordination game are maximising their individual rewards, conditioned on goals that may differ and without access to the observation, goal, action and reward of others. The agents may successfully communicate, in the sense that they coordinate their goals, but will not be rewarded if their policies cannot achieve them. Vice versa, they may succeed in an episode even if they don't communicate meaningfully.

3.2.7 Empirical results

Setup

We study the Dec-IMSAP in the Cooperative landmarks environment that we implemented using Simple Playgrounds [350]. This 2-D environment, illustrated in Figure 3.13, consists of a room with $L = 6$ landmarks on its walls and two agents that receive continuous-valued observations about the distance and angle to all landmarks and other agents. They can move by performing discrete-valued actions that control their angular velocity and longitudinal force. We consider navigation tasks where agents need to reach different landmarks and define goals as vectors of dimension L that are either one-hot or two-hot, the former corresponding to individual goals and the latter to cooperative. Formally, $g = [x_1, \cdots, x_l], \sum_l [x_l] \in [1, 2]$ where $x_l = 1$ indicates that landmark l needs to be reached by at least one agent for the goal to be achieved and landmarks are indexed starting from the white one and continuing clockwise (we provide the complete list of goal encodings in Appendix A.6.1, alongside a formal definition of the observation and action space). An episode finishes for an agent once it receives a reward or a time limit is reached. Then, it waits for others to also complete their episode before a new one starts. We illustrate an example in Figure 3.13: if the blue agent samples goal $[100000]$ and the green $[100100]$, then: a) if blue navigates to the white landmark and green navigates to the purple landmark the episode succeeds for both agents b) if blue navigates to the blue landmark and green to the white landmark then the episode succeeds for the blue agent. Each agent learns a goal-conditioned policy using PPO with a feedforward policy and uniform goal-sampling distribution $\mathscr{D}(\mathscr{G})$ (We provide the values of all agent hyper-parameters in Appendix A.6.2). To investigate the effect of using more complex intrinsic motivation mechanisms, in Appendix A.6.5 we replace uniform sampling with learning progress [250]. We have also studied a baseline that uses a recurrent policy in Appendix A.6.5 to investigate whether memory can help agents coordinate.

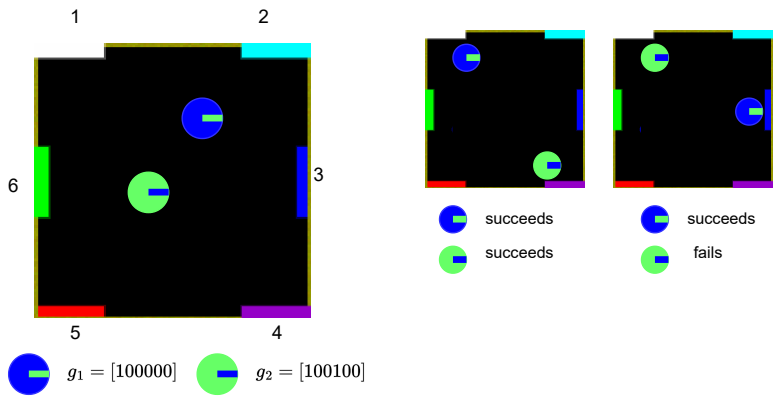


Figure 3.13: The Cooperative landmarks environment consists of a room with two agents and six landmarks, indicated as colored rectangles. Tasks are formulated as landmarks the agents need to navigate to. During training, the agents sample their own goals that can be individual or cooperative, as the ones chosen by the blue and green agents respectively. An episode may succeed for one agent and fail for the other, the outcome depending on the actions of both.

We introduce hyper-parameter β for controlling the relative importance between independent and cooperative goals by dividing rewards for individual goals by β. We do so to model the benefits of cooperation: outcomes that require cooperation often bring larger rewards than outcomes easily solved by a single agent (for example catching a big animal is more rewarding than catching a small one [383]). We set $\beta = 2$ here and study the effect of this hyper-parameter in Appendix A.6.5. To study the effect of environmental complexity, we performed experiments with a smaller environment ($L = 3$) in Appendix A.6.5.

In Section 3.2.7 we evaluate the role of goal alignment by designing baseline goal-sampling strategies for different levels of it. For a given $x\%$ desired level of alignment, each agent samples its own goal using $\mathscr{D}(\mathscr{G})$, but in $x\%$ of the trials we interfere in the sampling procedure and externally provide the agents with the same goal. Therefore, 0% alignment corresponds to autotelic agents sampling their own goals independently of the other at each episode and 100% alignment corresponds to a centralized goal-selection mechanism where a a goal is first sampled externally at the start of each episode and then provided to both agents (similar to the method used by [359]). We evaluate 0%-aligned (also referred to as independent), 50%-aligned and 100%-aligned (also referred to as centralized). We also evaluate a common method in MARL that follows the centralized training with decentralized execution paradigm (CTDE) [372], where every critic has access to all goals, actions and observations of the group. We refer readers to Appendix A.6.3 for an illustration of how these methods differ in terms of the information available to each agent.

Finally, in Section 3.2.7 we evaluate the ability of our proposed algorithm, the Goal-coordination game to reach the performance of the centralized baseline and provide insights into how alignment and performance co-evolve.

The role of alignment

We have hypothesized that agents not aligning their goals during training will not master cooperative goals, as they will collectively pursue them rarely and receive a noisy training signal as the goals

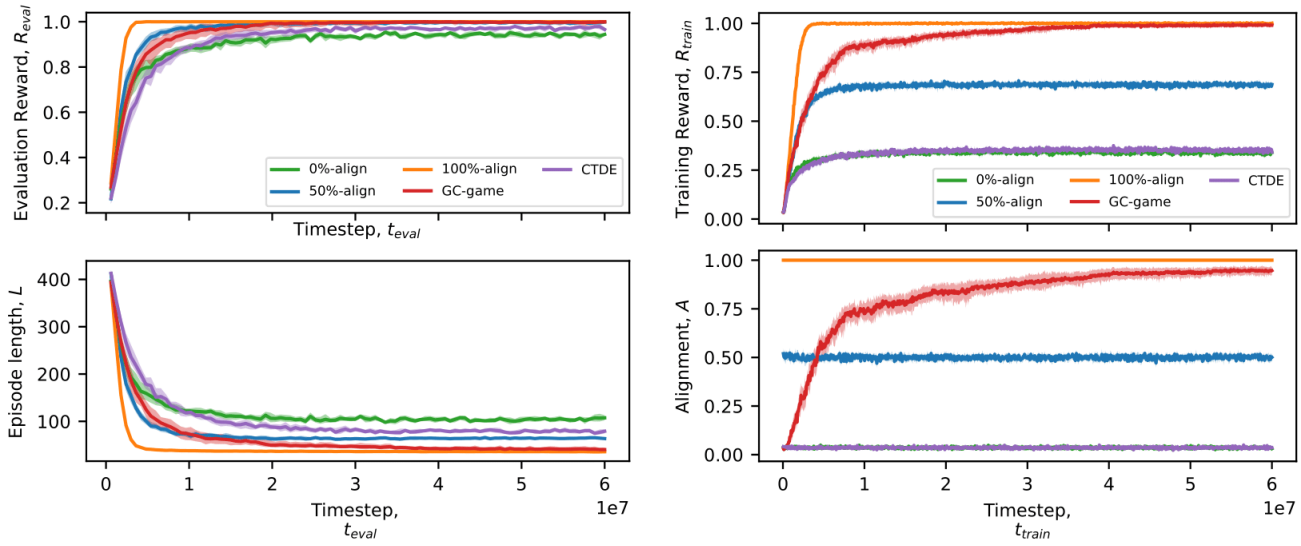


Figure 3.14: Comparison of baselines with different levels of alignment and the Goal-coordination game in terms of performance during evaluation (Left) and during training (Right). We present IQM values with stratified bootstrap confidence intervals computed over 20 seeds.

of others are not directly observable. We now examine this hypothesis by comparing the performance during evaluation and training trials between groups of centralized, independent and 50%-aligned agents in Figure 3.14. In addition to the collected rewards we monitor alignment during training trials and the length of the episode during evaluation trials, where shorter episodes indicate that the group solved the tasks quicker. We observe that ensuring alignment during training improves performance during evaluation. In particular, the evaluation reward at the end of training is 0.8277 ± 0.0436 for independent, 0.9133 ± 0.0027 for 50%-aligned and 0.9166 for centralized. Similarly for the episode length, independent requires significantly more time than other methods. Our study of the smaller environment in Appendix A.6.5 showed qualitatively similar behaviors, with differences between methods being less pronounced. Thus, alignment acquires more significance as the environment becomes more complex. The differences in performances are primarily due to cooperative goals; lowering alignment does not have a big impact on the individual goals (we confirm this in Appendix A.6.5, where we train and evaluate only with cooperative goals and observe similar conclusions as in the current setup but with more visible gap between methods).

As we discussed in Section 3.2.6, independent agents may fail because: a) as they cannot observe the goals of others and may choose incompatible goals during a training episode, the reward signal does not allow to discriminate between an infeasible episode and a feasible episode where the agents acted sub-optimally b) a large part of the training episodes is infeasible so the agents require more training time compared to centralized. To find out which of the two is the case we evaluated an additional method that we refer to as "both-goals": agents sample their goals independently but we provide both of them to each agent. In this way, the agent can learn which combinations of goals are incompatible and, thus, denoise the training signal. Our experiments showed that the both-goals method manages

to detect incompatible goals but still performs similarly to independent (we discuss this result in more detail in Appendix A.6.5). This suggests that the reason why independent fails is the large number of infeasible episodes and agrees with our observation that, when we decrease the number of goals in the environment, which dramatically reduces the probability of infeasible episodes, independent reaches the performance of centralized (see analysis in the environment with 3 landmarks in Appendix A.6.5). A similar behavior to both-goals is observed for the CTDE baseline, which achieves a slightly better performance. As we explain in our analysis of this method in Appendix A.6.5, including both goals in the value function enabled both CTDE and both-goals to detect infeasible episodes. Under CTDE, agents also exhibited more intra-episode adaptation, which may explain their superior performance. The baseline with the recurrent policy, analyzed in Appendix A.6.5, faces the same limitation and is more sensitive to the noise introduced by infeasible episodes compared to feedforward policies.

We should note that alignment is not sufficient for acting optimally in our environment as, even if both agents choose the same cooperative goal they still need to coordinate on who goes where. How can they do so with perfect success rate? We hypothesize that the agents will find it challenging to adapt to the other's behavior due to the high level of partial observability in the environment: without a recurrent policy and without observing the direction an agent is moving to, inferring the sub-goal pursued by the other is difficult. Instead, a specialization strategy where the two agents reach an agreement during training on who goes where (e.g. one agent always goes to the left-most landmark and the other to the rightmost) requires less effort. To detect this behavior, we search for specialization, i.e., policies that, when assigned with a cooperative goal during evaluation, are biased to one of its landmarks. We quantitavely measure specialization as the ratio of the episodes in which the agent went to its preferred landmark when following a cooperative goal. For example, if for goal $[101000]$ an agent went 7 times to $[100000]$ and 3 times to $[00100]$ this score would be 0.7. We observed that specialization correlates with alignment: independent specializes by 0.72 ± 0.0452, 50%-align by 0.8066 ± 0.0537 and centralized by 0.92 ± 0.083 (see Figure A.33 in Appendix A.6.5 for an illustration of these results) .

Learning to align goals

We have established that alignment is an efficient strategy for solving the Dec-IMSAP. To investigate whether it can be achieved without introducing centralization, we now turn to the evaluation of our proposed method for coordinating goals through communication, the Goal-coordination game, that we described in Section 3.2.6. We observe that,in Figure 3.14, the evaluation reward for the Goal-coordination game at the end of training is 0.9144 ± 0.0044. We also observe that early in training (time step $66 \cdot 10^5$) the Goal-coordination game collects less rewards than centralized. Similarly for the episode length, the Goal-coordination game is initially slower than centralized but at the end of training reaches its speed and surpasses independent and

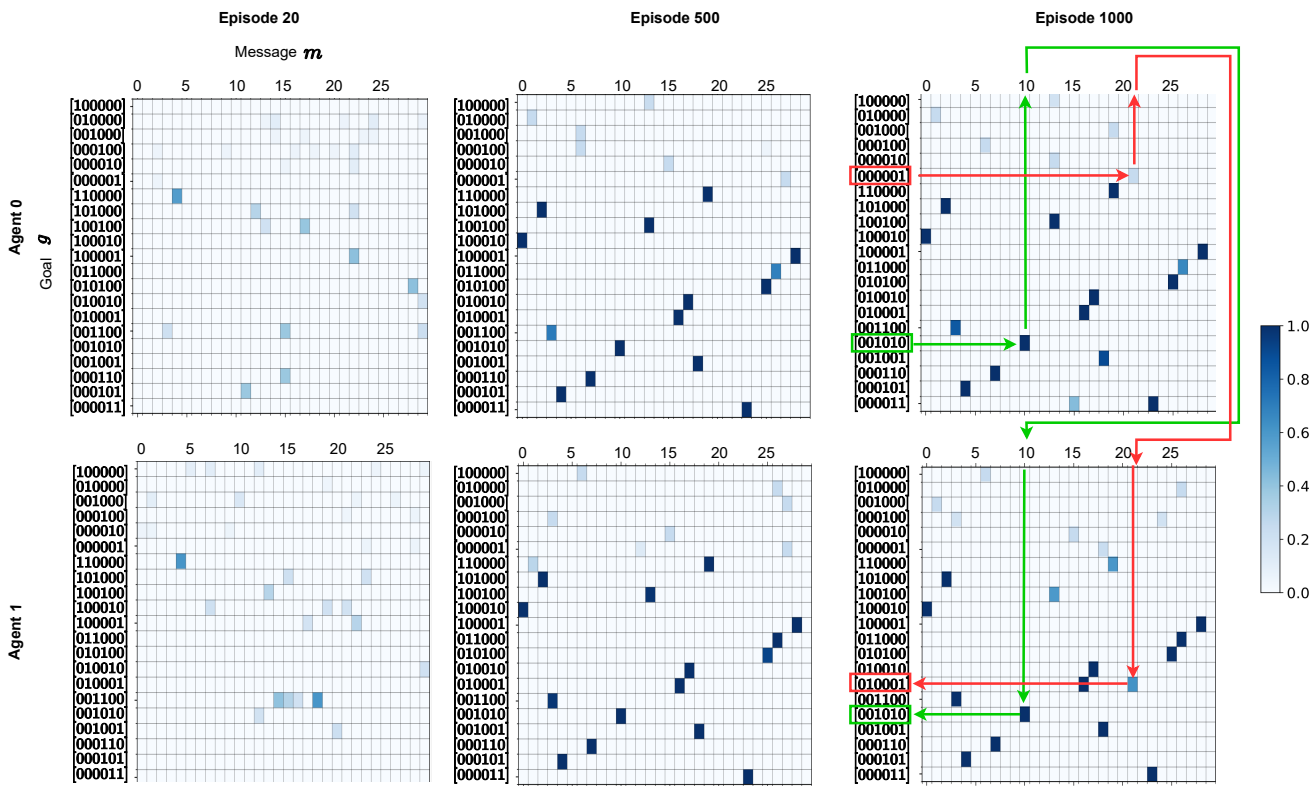


Figure 3.15: Evolution of the matrices of the Goal-coordination game early (left), in the middle (middle) and at the end of training (right): Rows correspond to goals, with individual goals assigned to the first 6 rows, columns correspond to messages and the intensity of a cell indicates the confidence in a goal-message association. The green arrow highlights communication that leads to goal alignment (both agents pursue goal [001010]) and the red communication that leads to the "risky follower" behavior (agent 0 pursues goal [000001] while agent 1 pursues goal [010001]).

50%-align. Next, we take a deeper look at its dynamics to understand these behaviors. In particular, we study the update matrices at early and later stages of training to understand why performance starts off bad but then reaches the optimal value.

In Figure 3.15, we visualize the matrices for a simulation that differs from the one in Figure 3.14 only in that β is increased from 2 to 4. As we discuss in Appendix A.6.5, increasing β does not affect performance but leads to interesting emerging behaviors that our study of the matrices can reveal. As we described in Section 3.2.6, each agent is equipped with a matrix mapping messages to goals and updates its cells to maximize its individual rewards during training episodes. The rows of the matrices correspond to goals and the columns to messages. We make the convention here of plotting the individual goals first, so the first 6 rows correspond to individual and the following 15 to cooperative goals. We have set the message size to a slightly higher value than the number of goals, i.e., $M = 30$. As we show in Appendix A.6.5 having more messages than goals facilitates training by decreasing the probability that the matrix updating will get stuck. Rows and columns where we can find a single cell with higher intensity than others indicate a converged goal-message association. We can detect alignment by tracing if the goal-message associations of the two agents agree. We observe that, early in training, the agents have low confidence for most associations and alignment has not been achieved. By the middle of training, however, the two matri-

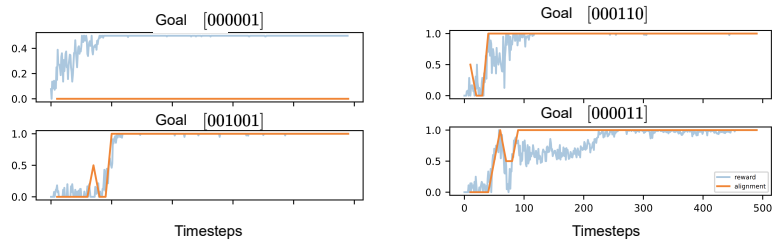


Figure 3.16: Co-evolution of alignment and rewards for different goals during training using the Goal-coordination game. The top-left plot corresponds to an individual goal and the rest to cooperative ones.

ces are almost identical (follow the green arrow for an example of communication leading to aligned goals). Looking at the matrices in Figure 3.15 we can see that not all goals are aligned. This is because in some cases (see the red line for an example) the leader samples an individual goal that the follower interprets as a cooperative one. This "risky" behavior is useful for the follower, as cooperative goals are more rewarding than individual ones. If the message received by the follower convinces it that the leader is pursuing an individual goal, the follower might have interest to pursue a cooperative goal compatible with it, which will lead to maximum reward. In a daily life analogy, if your housemate tells you they will buy pasta for tonight (their individual goal), you may buy pasta sauce (a cooperative goal based on your expectation that the other will fulfill its individual goal) instead of rice (a different individual goal).

A challenging feature of the Goal-coordination game is that the matrices and policies are updated simultaneously. This can lead to a chicken-and-egg problem: the matrix updates may fail even if the goal-message association is correct because the policy has not managed to solve a goal. Or the policy may struggle to solve goals because of bad goal-message associations that lead to episodes infeasible to solve. In Figure 3.16 we monitor the co-evolution of alignment and rewards during training for a random subset of the goals. We observe that, for cooperative goals, rewards and alignment are highly correlated with improvements in one driving improvements in the other, while, for individual goals, rewards are maximized without requiring alignment.

3.2.8 Discussion

We present a new problem for formalizing intrinsically-motivated multi-agent goal exploration in a decentralized training paradigm, Dec-IMSAP, and propose an algorithm for solving it, the Goal-coordination game. We empirically observe that shared intentionality, which we measure as alignment of cooperative goals during training, plays an important role in a group's ability to solve a wide diversity of tasks. Aligned agents do not only get the highest rewards but also do so quickly. We also show that, under the Goal-coordination game, alignment emerges without being explicitly rewarded and groups reach equal performance to a centralized setting that guarantees alignment. We observed that groups with higher alignment solve the tasks by specializing instead of monitoring and adapting to others, which, as has

been observed in previous MARL studies [152], is a behavior challenging to emerge unless explicitly rewarded.

We have adopted a descriptive rather than normative approach, common in the study of open-ended learning [9, 39]. Our aim was to get groups of agents that learn a maximally diverse behavioral repertoire, but observed the emergence of behaviors such as the risky follower. Whether such behaviors are desirable or not, depends on the application at hand.

Our study of the Goal-coordination game is limited to populations of two agents and discrete message and goal spaces. Extending it to larger groups is important for scaling up its applicability. We hypothesize that in such settings specialization will no longer lead to optimal performance and that the goal-conditioned policy will need to be extended by conditioning it on messages and introducing recurrency to equip agents with memory [183]. Also, having shown that increasing environmental complexity increases the importance of alignment (by comparing environments with different numbers of landmakrs), we believe that an interesting extension of this work would be to test our approach in a more complex, multi-agent environment like Grafter [384]. To extend the Goal-coordination game to continuous message and goal spaces, we can adopt approaches based on energy-based models employed in previous works [385]. Finally, while our empirical study considers a pre-defined goal space, we should note that this is not necessary for autotelic agents who can in general learn their own goal representation [144]. We envision studies of the Goal-coordination game where both goals and messages emerge (to study for example language evolution [386, 387]).

We believe that the Dec-IMSAP, can be of interest in real-world scenarios such as robotics for disaster rescue or extraterrestrial exploration. It allows to consider a population of goal-conditioned RL agents that learn how to achieve a wide diversity of cooperative tasks in a fully-autonomous manner. In this way, a user could place agents (simulated or robotics) in some environment and let them interact without any supervision for a period of time. At the end of this training phase, the agent population will have autonomously learned how to achieve diverse individual and collaborative goals without any supervision and a human user will be able to benefit from these acquired skills.

3.3 Chapter conclusion

In this chapter, we explored how efficient cooperative exploration mechanisms can be meta-learned as a response to variability in a large distribution of environments.

In the first section, we examined the emergence of collective exploration strategies in decentralized learning agents through meta-training on procedurally generated task sequences. Our results reveal intriguing generalization capabilities, with agents adapting to longer tasks, novel objects, and unseen dynamics beyond their training distribution. Finally, though this would require further work to be confirmed,

Summary

- Meta-training on a diversity of procedurally generated multi-step hierarchical tasks enable the emergence of collective exploration strategies in group of decentralized agents.
- Introducing communication in group of autotelic agents enable to align their goals, improving the efficiency of cooperative learning and exploration.

the meta-learned agents seem to display a proto sub-goal selection mechanism – akin to autotelic agents – switching what they explore during the episode.

In the second section, building on these results, we explored how communication could help a group of autotelic agents to explore more efficiently by aligning their goals. We first showed that agents that don't align their goals (sampling them randomly) learn less efficiently. We then introduced a communication mechanism and decentralized training algorithm that result in "emergent shared intentionality" from the learning of a communication protocol being driven only by the maximization of their own individual reward. Such emergent shared intentionality enables agents to align their respective goals, resulting in more efficient exploration and training.

A natural extension of this work would involve integrating communication into the framework from the first section (Sec.3.1) and analyzing the meta-learned communication strategies. This could reveal communication as a tool for sharing intentions, as seen in our second contribution (Sec.3.2). Additionally, agents might use communication to exchange information about their discoveries – for instance, sharing details about functional combinations of objects –potentially paving the way for proto-culture to emerge.

In fact, communication could also be used beyond mere goal alignment to encompass richer forms of information exchange during an episode. For example, agents might share detailed strategies for solving tasks or collaborate more efficiently through nuanced signals. Ideally, the communication would contain as little bias as possible (i.e. no explicit goal etc.), but rather in a "free form" – such as continuous signals or continuous drawing [388]. This unstructured "free form" communication could enable the agents to potentially meta-learn rich compositional signals or even a common complete "language".

While efficient exploration behaviors emerged in sec.3.1, they were always incentivized by the agents' expectation of a potential direct reward from trying out some object affordances. In fact, we did not seem to observe any exploration behavior that indicates truly curious behavior – for example, behaviors only aimed to gain useful knowledge on the rules of the current task without any direct reward for it. More generally, further work could explore the meta-learning of more general complex exploration strategies. We report in Sec.4.4 preliminary work in this direction, showing the meta-learning of emergent curious behavior directed toward gaining information without any direct reward associated with it.

Additional Papers and code. 4

4.1 Emergent kin selection of altruistic feeding via non-episodic neuroevolution

Context

This contribution is the result of the visit from Max Taylor-Davies (School of Informatics, University of Edinburgh, Edinburgh, Scotland) in the FLOWERS team. It is based on the eco-evolutionary system introduced in Sec.2.1.

- Taylor-Davies, M., Hamon, G., Boulet, T., Moulin-Frier, C. (2024). *Emergent kin selection of altruistic feeding via non-episodic neuroevolution.* arXiv preprint arXiv:2411.10536. Paper, Code

This paper got a long oral presentation at the International Conference on the Applications of Evolutionary Computation – evoApps – 2025.

Kin selection theory has proven to be a popular and widely accepted account of how altruistic behaviour can evolve under natural selection. Hamilton's rule, first published in 1964, has since been experimentally validated across a range of different species and social behaviours. In contrast to this large body of work in natural populations, however, there has been relatively little study of kin selection *in silico*. In the current work, we offer what is to our knowledge the first demonstration of kin selection emerging naturally within a population of agents undergoing continuous neuroevolution. Specifically, we find that zero-sum transfer of resources from parents to their infant offspring evolves through kin selection in environments where it is hard for offspring to survive alone. In an additional experiment, we show that kin selection in our simulations relies on a combination of kin recognition and population viscosity. We believe that our work may contribute to the understanding of kin selection in minimal evolutionary systems, without explicit notions of genes and fitness maximisation.

4.2 Evolving Reservoirs for Meta Reinforcement Learning

Context

This contribution is the result of the master internship of Corentin Léger co-supervised by members of the FLOWERS team (Clément Moulin-Frier, Eleni Nisioti and me) and the MNEMOSYNE inria team (Xavier Hinaut) at Inria.

- Léger, C., Hamon, G., Nisioti, E., Hinaut, X., Moulin-Frier, C. (2024). *Evolving Reservoirs for Meta Reinforcement Learning.* **In International Conference on the Applications of Evolutionary Computation** (Part of EvoStar) (pp. 36-60). Cham: Springer Nature Switzerland.
  Paper, Preprint, code

This paper was presented at the International Conference on the Applications of Evolutionary Computation (Part of EvoStar) 2024.

Animals often demonstrate a remarkable ability to adapt to their environments during their lifetime. They do so partly due to the evolution of morphological and neural structures. These structures capture features of environments shared between generations to bias and speed up lifetime learning. In this work, we propose a computational model for studying a mechanism that can enable such a process. We adopt a computational framework based on meta reinforcement learning as a model of the interplay between evolution and development. At the evolutionary scale, we evolve reservoirs, a family of recurrent neural networks that differ from conventional networks in that one optimizes not the synaptic weights, but hyperparameters controlling macro-level properties of the resulting network architecture. At the developmental scale, we employ these evolved reservoirs to facilitate the learning of a behavioral policy through Reinforcement Learning (RL). Within an RL agent, a reservoir encodes the environment state before providing it to an action policy. We evaluate our approach on several 2D and 3D simulated environments. Our results show that the evolution of reservoirs can improve the learning of diverse challenging tasks. We study in particular three hypotheses: the use of an architecture combining reservoirs and reinforcement learning could enable (1) solving tasks with partial observability, (2) generating oscillatory dynamics that facilitate the learning of locomotion tasks, and (3) facilitating the generalization of learned behaviors to new tasks unknown during the evolution phase.

4.3 Open-source implementation of a transformer-XL based RL agent.

We provide an open-source JAX [289] implementation of TransformerXL [389] based agents trained with proximal policy optimization (PPO [320]) in a reinforcement learning setup following the details of : "Stabilizing Transformers for Reinforcement Learning" from Parisotto et al. [231]. The implementation is accessible at this link.

This code was used in our contribution Sec.2.2 and Sec.4.4.

We also report results on the challenging craftax RL environment [232]. In particular, without much hyperparameter search, our implementation beats the baseline presented in the original craftax paper, including recurrent neural networks (RNN) based agents trained with PPO (Fig.4.1). With a budget of 1e9 timesteps, we achieve over 3 seeds a mean normalized return of 18.3% compared to 15.3% for PPO-RNN according to the craftax paper. Notably, the implementation reaches

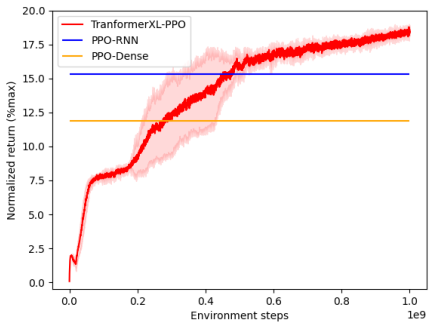


Figure 4.1: Learning curve on craftax [232] of our tranformerXL implementation (red) compared to baselines reported in the paper: recurrent neural network (blue) and linear neural network (yellow).

the 3rd level (the sewer) and obtains several advanced advancements, which were both not achieved by the methods presented in the paper even when trained for ten times more interactions. In fact, when trained over 10e9 timesteps, the RNN based agent achieved a mean normalized return of less than 17.5%.

With a budget of 4e9 timesteps, our implementation achieves a normalized return of 20.6 %, visits the 3rd floor (the sewer) a decent amount of time and achieves several advanced achievements. We report in appendix.A.7 the achievements success rate along training. The instructions to reproduce these results are available in the code repository.

The implementation takes advantage of JAX parallelization to speed up training. The training of a 5 million parameters transformer on craftax for 1e9 steps (with 1024 environments in parallel) takes about 6h30 on a single nvidia A100 GPU. The implementation also **supports multi-GPU training**, achieving even higher speed.

4.4 Meta-learning curiosity through reward maximization in a variable compositional environment.

Context

We report in this section preliminary work conducted during the thesis about the meta-learning of curious behavior. In particular, this work is a direct follow-up of Sec.3.1, where we explore further how additional variation in the environment can potentially lead to curious information-seeking behavior.

As shown in our contribution Sec.3.1 and [184], agents can meta-learn to explore the environment. However, in those works, the meta-learned exploratory behavior is directed towards actions that might give some reward or get the state of the environment closer to getting a reward. In particular, they do not seem to display exploration actions that are done **only** to get information and are not potentially rewarding (nor bring the state of the environment closer to getting a reward). In this work, we call "**information-seeking behavior**" such exploration that only aims at collecting information about the environment dynamics.

Some works in meta RL displayed such information-seeking behavior but often prespecified an exploration phase, often with its own exploration policy separated from the exploitation policy [333, 334] (and even in some cases [334] directly optimize the exploration policy to maximize information gain on the task). However, prespecifying a fixed exploration period prevents the agent from autonomously learning when to switch from exploration to exploitation and instead imposes such a constraint on the task. In particular, this type of technique might not be possibly applied to non-episodic continuous episodes where there is often not a concrete notion of the beginning of the task (but rather a continuous stream of new challenges and

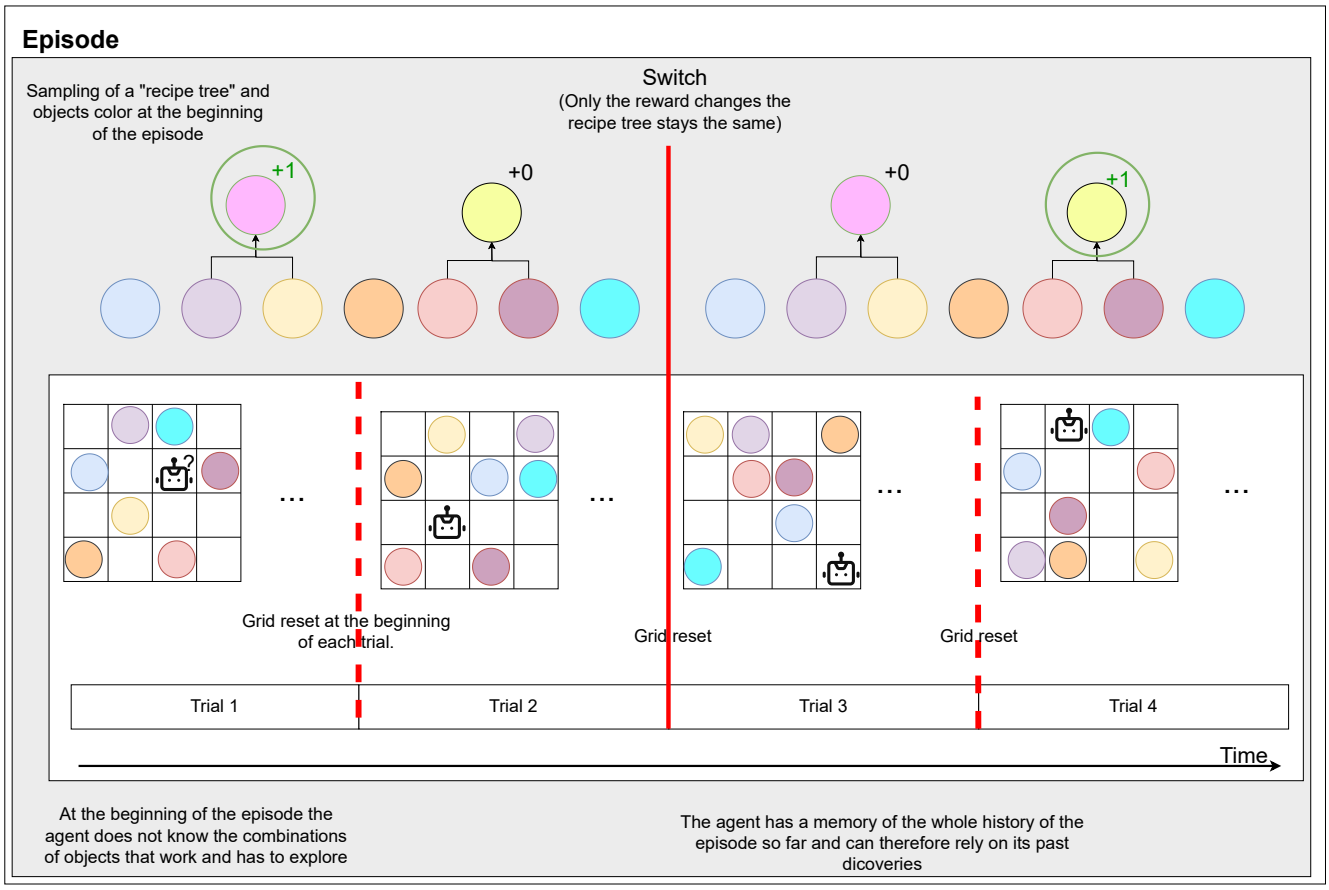


Figure 4.2: Overview of an episode of the "recipe environment". We follow a multi-trial episode framework as is common in meta-RL [182–184]. At the beginning of the episode the agent does not know anything about the recipe tree and has to explore. The agent can use its memory component to also exploit past discoveries as the recipe tree does not change across the episode. We only add an additional switch of reward location to add variability in the environment.

opportunities often with a blurry boundary between them). In addition, more complex environments and strategies might require an adaptive switch of behavior between exploration and exploitation.

The objective of this work is to meta-learn information seeking exploration strategies without prespecifying an exploration period.

4.4.1 Description of the task

The agent meta-learns a transformer-based policy, i.e. a policy mapping the full history of (observation, action, reward) tuples since the start of the episode, to action. The agent is placed in a *recipe environment* (Fig.4.2): a grid world with diverse objects of different colors that can be picked and placed, as well as combined to create new objects. The rule specifying how two objects can be combined to create a third one is called a *recipe* (in the form $A+B \rightarrow C$). Multiple recipes can be combined to form a *recipe tree*, as illustrated at the top of Fig.4.2. Some recipes are rewarding while others are not. The recipe environment, therefore, induces two different exploration challenges. The first challenge consists in exploring a fixed recipe tree in order to learn its structure (i.e. the set of underlying rules). The second

challenge consists in finding which recipe is rewarding within a given recipe tree.

In our experiments, at each episode, we first sample the colors of the objects, a random recipe tree and a rewarding recipe and let the agent interact with the environment for some time. At some point within the episode, we switch the rewarding recipe, while keeping the same recipe tree (Fig.4.2). Our objective is to study which type of exploration strategy is learned in such a variable environment. In particular, once the agent has found the rewarding recipe in the first phase (i.e. solved the second challenge explained above), it might have interest to continue seeking information about the recipe tree structure (i.e. first challenge) in order to anticipate the switch to another rewarding recipe.

We follow the multi-trial episode setting as is common in meta-RL [182–184] with environment reset but memory propagation. We let the agent meta-learn when to explore adaptively.

During a trial, we follow the reward structure used in [184], where the agent gets a reward until the end of the trial when it has discovered the rewarding recipe. Even though the agent has found the rewarding recipe, for the rest of the trial the agent can still continue to interact with the environment (and will continue to get the reward whatever it does), potentially to explore further.

At the beginning of the episode, the agent does not know anything about the recipe tree and has to explore, similar to our contribution Sec.3.1. Taking advantage of its memory, the agent can meta-learn to memorize the "working" recipes and exploit them.

The agent observes the reward and therefore can know which recipe is rewarding. We also add in the observation a binary flag telling if the combination that was just tried is successful or not (potentially helping the attention-based memory to spot useful observations in the history).

Overall, the training setup is very similar to [184], but in a simpler toy compositional environment and, most importantly, with an intra-episode variability. Compared to Sec.3.1, we use a gridworld environment, removing the potentially hard sensorimotor aspect of the task (and we add variability during the episode).

We use our open-source code for transformerXL with PPPO (Sec.4.3).

4.4.2 Results

We meta-train the agent on a diversity of generated recipe trees and object colors and report the behavior of the agent tested on an episode. In particular, we measure the cumulative amount of time the agent has tried each combination of objects, indicating exploration and exploitation capabilities (for example, if the agent tries several times a non-working recipe). We also measure the reward obtained by the agent.

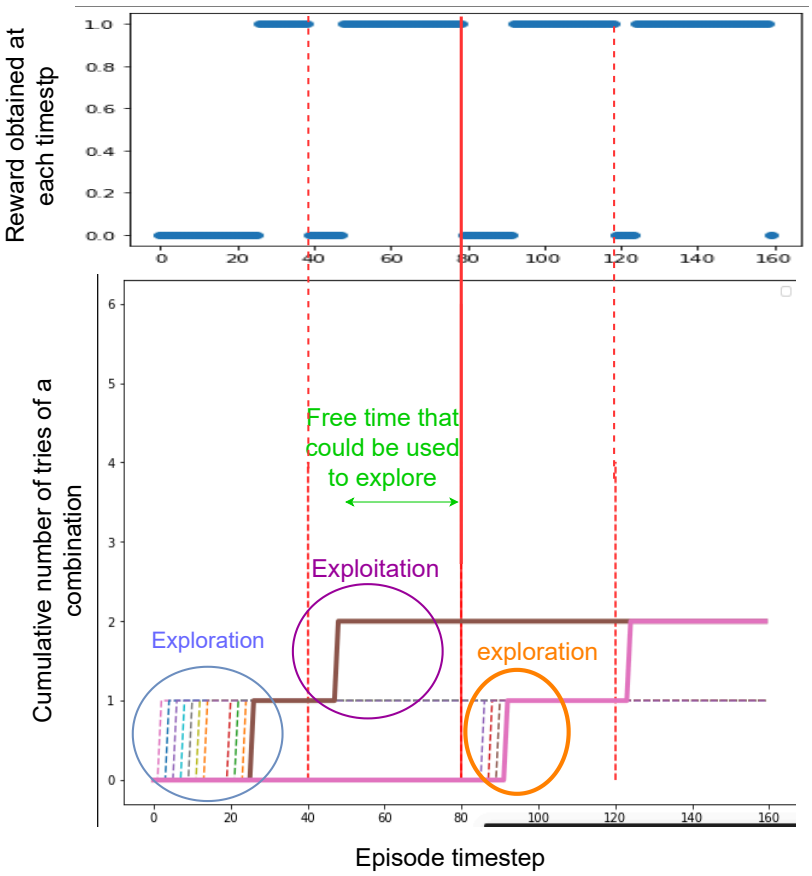


Figure 4.3: **Test episode of an agent that did not meta-learn to perform curious information-seeking behavior.** We measure the cumulative number of trials of each combination of objects, the dotted lines are the "non working" combinations while the plain lines are the "working" combinations. We also report the reward obtained at each time step (top). The agent effectively learns to explore (blue) and exploit (purple) but does not use its "free time" to anticipate the change (green) and therefore has to explore at the beginning of trial 3 (orange) loosing potential reward. The first exploration phase (blue) is not considered an information-seeking behavior as each try of combination might potentially give reward.

First results without information seeking behavior. We report in Fig.4.3 results with an agent trained with an infinite number of possible colors for the object; where curious information-seeking behavior does not emerge. This "failed" result is interesting to understand the task and the behavior we expect. In fact, we see that the agent effectively learns to explore (as in our contribution Sec.3.1) and also to memorize and exploit its discovery in later trials as in [184]. In fact, the use of the memory allows the agent to directly combine the rewarding recipe it has already found in the previous trials without having to explore again.

However, after the switch of recipe, the agent explores again to find the new recipe that is rewarding (orange in Fig.4.3). This is suboptimal as the agent loses time and thus reward exploring again, while it could have used its "free" time (green in Fig.4.3) during previous trials (as the recipes do not change across the switch and the agent has already found the reward and therefore cannot get more).

The first exploration phase (blue in Fig.4.3) is not considered an information-seeking behavior as each try of combination might potentially give a reward.

Emergence of information seeking behavior. When trained with a finite number of possible colors (which might help the agent to build a better representation), we observe in Fig.4.4 the emergence of information-seeking behavior. In fact, the agent anticipates the change and ex-

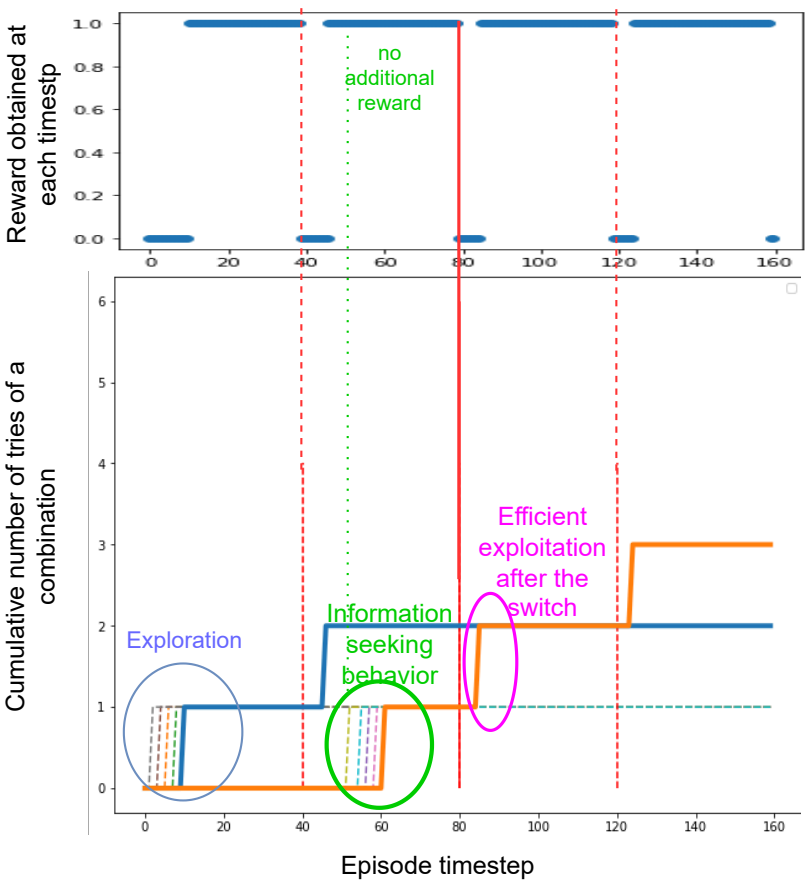


Figure 4.4: **Test episode of an agent that meta-learned to perform curious information-seeking behavior.** We measure the cumulative number of trials of each combination of objects, the dotted lines are the "non working" combinations while the plain lines are the "working" combinations. We also report the reward obtained at each time step (top). The agent effectively use its free time to perform "information seeking behavior" (green) as this action does not give any reward but allow the agent to gain information that is useful to anticipate the switch and exploit efficiently just after (orange).

plores during its "free time", allowing it to directly exploit just after the switch (compared to the previous result Fig.4.3 which lost reward to explore again). In particular, this exploration behavior (highlighted in green in Fig.4.4) is always done after completing the rewarding recipe in trial 2, i.e. in a phase when it's not possible to get any additional reward. The agent therefore effectively performs explorative actions without getting any reward from them. However, this enables information seeking about the general recipe tree structure, that can be memorized and reused later on. Indeed, we see (pink in 4.4) that the agent effectively exploits the information gained during its curious information-seeking behavior.

Note that in both training setups presented above, the agent might stumble on the non-rewarding recipes while searching for the rewarding one (before finding it) in the first exploration phase. In this case, the agent memorizes it and does not have to rely on the information-seeking behavior, which is the reason why we only showed episodes displaying the other case.

These preliminary results show again the importance of environmental variability for the emergence of complex exploratory behavior. In particular, we elicit a simple toy training paradigm enabling an efficient use of "free time" and leading to the meta-learning of curious information-seeking behavior.

Further work could explore more variability in the episode (e.g. with a switch happening randomly), as well as more general environmental

dynamics to explore by the agent. This could lead to more general curiosity behavior. In particular, once information-seeking behavior has emerged it might be easier to transfer it to other harder tasks. Therefore, introducing a curriculum of tasks could be an interesting future direction, potentially ultimately leading to an agent capable of open-ended curious exploration of the environment (constantly exploring new possibilities). Further works could also explore how similar behavior could emerge from evolution, for example using fitness taking into account satiety which has a similar structure as the reward used here (with satiety leading to no more ”reward” from eating more).

Discussions 5

5.1 Summary of the thesis

As explained in the introduction, the goal of this thesis was to explore mechanisms promoting open-ended dynamics in artificial systems. In particular, we focused on the interactions between adaptive agents and the environments, with a special emphasis on the feedback loop they might induce, potentially leading to never-ending new adaptations. The thesis studied how diverse environmental dynamics and adaptive mechanisms, operating at multiple spatio-temporal scales, can induce interesting phase transitions at the system level.

In chapter 1, we explored the emergence of individuality and proto-evolutionary dynamics in an initially lifeless environment in the continuous cellular automata Lenia.

- In a first contribution Sec.1.2, using machine learning techniques such as gradient descent, diversity search and curriculum learning we explored the parameters space of Lenia to find rules leading to the systematic self-organization of macro-structures with sensorimotor capabilities. In particular, the self-organized macro individuals reacted to perturbations and changed direction without any central brain to take decision but rather from the collective self-organization of its simple constituents. Finally, we observed interesting generalization capabilities to conditions not seen during the search (such as different obstacle shapes, new scale, noise, or multi-agent simulations) showing promises for robust morpho-cognitive agents.
- In a second contribution Sec.1.3, we introduced an extension of Lenia named Flow Lenia which added mass conservation in the system. Mass conservation made it easier to find localized patterns. Most importantly, the extension of the system allowed for the coexistence of several localized rules in the system, enabling "multi-species" simulations. In particular, the dynamic of the system in the multi-species case showed emergent intrinsic proto-evolutionary dynamic, from the physics of the system alone. This dynamic is highlighted by measuring evolutionary activity metrics and displaying phylogenetic trees.

The first chapter, therefore, introduced the emergence of individuality and adaptation mechanisms. From this point on, our contributions adopted a more classical setup featuring an embodied agent with a brain, interacting with a well-separated environment through predefined sensors and actuators. This change of paradigm, from a so-called enactivist framework to a mechanistic framework (Sec X), enabled to study in more detail the reciprocal interactions between adaptive agents and their environment in Chapters 2 and 3.

Quick summary chapter 1

- Applying diversity search algorithms to a continuous cellular automaton enables the discovery of artificial creatures displaying features of sensorimotor agency with interesting generalization abilities.
- Introducing mass conservation in a continuous cellular automata enables multi-species simulations bootstrapping a proto-evolutionary mechanism.

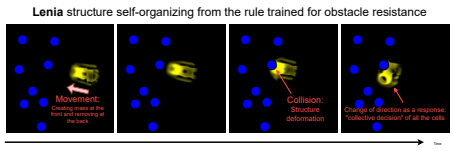


Figure 5.1: Emergent sensorimotor agency in contribution Sec.1.2

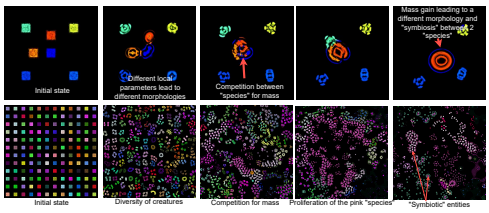


Figure 5.2: Emergent evolutionary dynamics in contribution Sec.1.3

In chapter 2, we investigated the impact of the adapting agents on the environment through niche construction. Niche construction ultimately alters environmental pressures, which in turn leads to new adaptations, ultimately leading to eco-evolutionary feedback effects.

Quick summary chapter 2

- Large scale experiments showing the important effects of eco-evolutionary feedbacks.
- Neuroevolution of efficient sustainable behavior through physiological reproduction, without any explicit objective being maximized.
- Different behavioral strategy coexisting, elicited by isolation and behavioral tests in "lab environments".
- Learning of collective eco-engineering strategies with the emergence of agriculture.
- Eliciting the conditions favoring the discovery of agriculture.

- In a first contribution Sec.2.1, we highlighted the importance of these eco-evolutionary dynamics by introducing continuous long non-episodic simulations with hundreds of neuroevolving agents foraging for spreading resources in a gridworld. We also adopted an energy based reproduction mechanism, where agents reproduce if their energy is above a certain threshold for some time. This minimal criterion based reproduction departs from traditional evolutionary algorithms maximizing an explicit objective or a fitness function. We reported strong coupling between agents and the environments, for example with the presence of Lotka-Volterra cycles. Interestingly, agents evolved sustainable collective behaviors where they foraged locally with parsimony allowing for the spreading of the resources. We hypothesized that such behaviors evolved as a way to optimally propagate a lineage, where future offsprings will benefit from a sustainable maintenance of the resource stock.
  A follow up of this contribution is introduced in Additional papers Sec.4.1, where we studied the emergence of altruistic behavior in similar embodied simulations. In this contribution, agents evolve altruistic feeding behavior towards their offspring, here again as a way to improve the propagation of their lineage – in line with kin selection theory.
- In a second contribution Sec.2.2, we explored more complex niche construction by studying the emergence of agriculture in groups of agents. In a gridworld with competing plants, we studied how a population of reinforcement learning agents collectively learn to promote the growth of a beneficial plant by spreading its seeds, watering it, and removing the unwanted plant. In particular, we explored the influence of both environmental factors (e.g. the parameters of the plants growth) and cognitive factors (e.g. how much the agent takes into account the future) favoring the discovery of agricultural practices in a multi-agent setting. We also highlighted specialization of the agents in some cases, with some of the agents watering the plants while some others planting seed (i.e. division of labor).

As highlighted throughout this chapter, niche construction and eco-evolutionary dynamics can lead to significant variability within the environment (for example, in the availability of resources). This high variability necessitates equally rapid adaptation mechanisms, enabling agents to adapt over shorter timescales. We focused in more detail on these aspects in chapter 3.

In chapter 3, we explored the emergence of learning and exploration as a response to environmental variability. In particular, we controlled the environmental variability and studied how groups of decentralized agents learn to collectively explore efficiently.

Quick summary chapter 3

- Meta-training on a diversity of procedurally generated multi-step hierarchical tasks enables the emergence of collective exploration strategies in group of decentralized agents.
- Introducing communication in group of autotelic agents enables goal alignment, improving the efficiency of cooperative learning and exploration.

- In a first contribution Sec.3.1 relying on meta-reinforcement learning techniques with agents equipped with recurrent architecures, we studied how a group of decentralized learning agents trained on a diversity of procedurally generated hierarchical task could meta-learn cooperative exploration strategies. We showed that, in this context, including single agent training episodes is efficient at mitigating issues with credit assignment that was otherwise leading to free-riding strategies. Interestingly, the meta-learned policies generalized to new settings not seen during training such as new objects, longer and more complex tasks and new tasks. Finally, the learned behavior seem to display a proto sub-goal selection mechanism switching what they explore during the episode.
- In a second contribution Sec.3.1, we pre-equipped agents with this autotelic (goal selection) mechanism and studied how communication could enhance the learning efficiency by aligning the agent's respective goals. In particular, we first showed that non-aligned goals (for example through random goal selection) lead to suboptimal behavior. We then introduced a communication mechanism and decentralized training algorithm that resulted in "emergent shared intentionality" [338] from the learning of a communication protocol being driven only by the maximization of their own individual reward. Such emergent shared intentionality indirectly enabled agents to align their respective goals, resulting in more efficient exploration and training.
- We introduced in additional paper section Sec.4.2 a work studying the meta-learning of parameters of a cognitive architecture allowing downstream fast adaptation to a distribution of task. We also introduced, in Sec.4.4, preliminary work on the meta-learning of information seeking exploratory behavior – exploration to gain information about the structure of the environment without any direct reward for it. This behavior allows to anticipate potential future changes in the environment during the lifetime of the agents.

Throughout this thesis, we explored emergent complexity and phase transitions across different scales. We embraced a bottom-up approach both in each and across contributions where the resulting dynamics from simulations were often used as the basis for the next ones to focus on more specific mechanisms in a controlled way. In particular, we went from the emergence of individuality and proto-evolutionary dynamics; to the study of complex agent-environment interactions with niche construction and eco-evolutionary feedbacks; to the emergence of learning, exploration (and communication). We recall in Fig.5.3 an overview of the general structure and logic of the thesis.

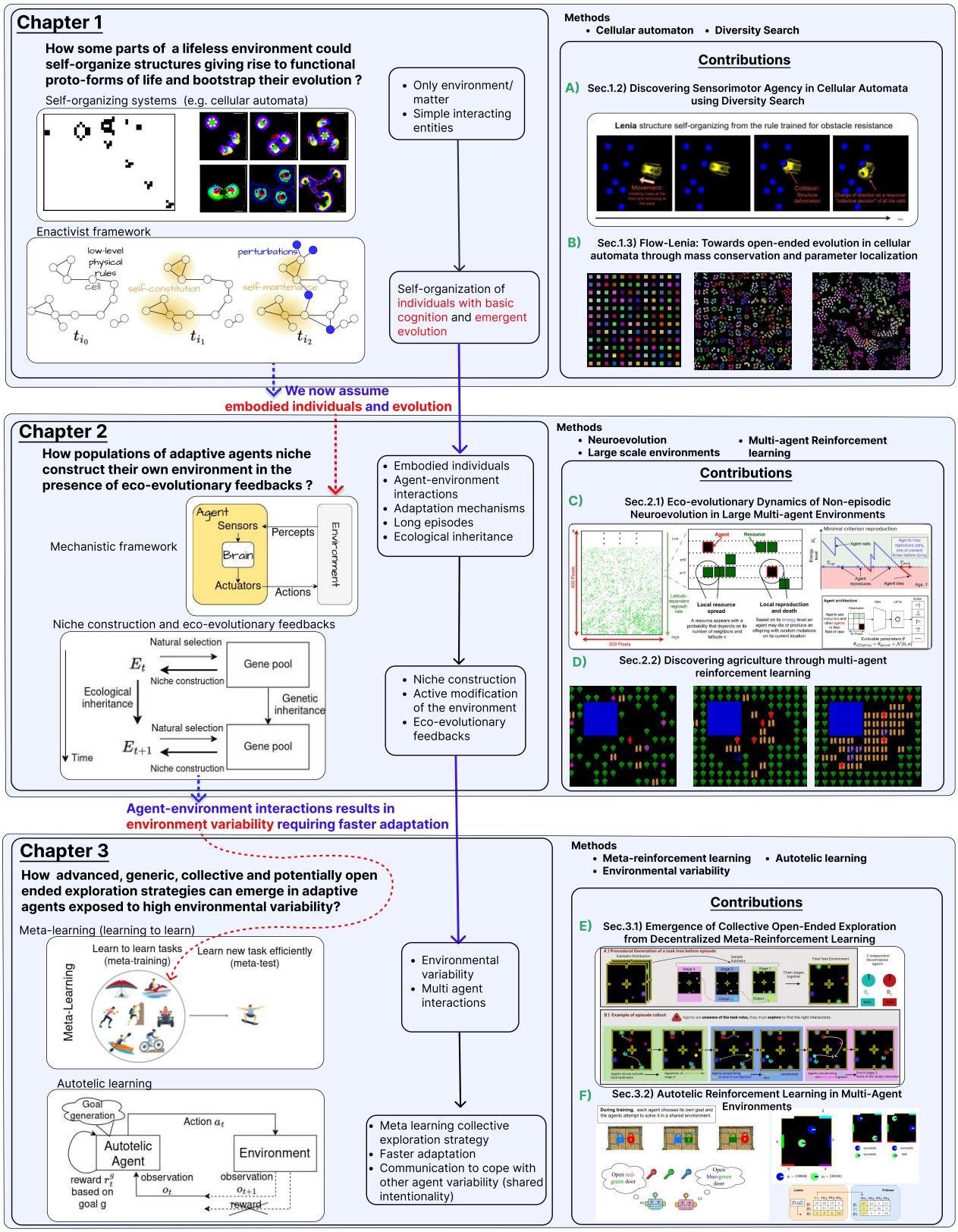


Figure 5.3: Overview of the contributions of the thesis. The thesis followed a progressive structure where the emergent behaviors derived in earlier chapters formed the basis (or initial state) for subsequent investigations.

5.2 Perspectives and limitations of the contributions.

For specific limitations and perspectives of individual contributions, we refer to the discussion sections within each corresponding section and chapter. We will now bring the pieces together in an attempt to synthesize a larger picture on how to design artificial systems featuring open-ended dynamics.

We start by discussing perspectives focusing on specific architectural components:

- Environment design, (Sec.5.2.1)
- Agent design, (Sec.5.2.2)
- Multi-agent interactions (Sec.5.2.3).

We then provide a general discussion on emergent complexity and engineered biases in Sec.5.3.1, the challenges of measuring open-endedness in Sec.5.3.2, and finally how our work could shed light on theories about the real world in Sec.5.3.3.

5.2.1 Environment design

The neural network architectures used in chapters 2 and 3 demonstrated complex adaptive capabilities across various scenarios. However, as we have seen, the complexity of evolved or learned behaviors strongly depends on the complexity of the environment in which adaptive agents operate. Similar results can be observed in other works showcasing the generalist capabilities of recurrent neural network-based architectures [44, 184, 390].

In this context, we consider that achieving generalist agents in AI is not only a problem of designing efficient cognitive architectures, but also –and perhaps more importantly– of designing environments whose dynamics are complex enough to support the open-ended adaptation of artificial agent populations.

In this section we will discuss what we think are the main building blocks of environments and training paradigms able to foster open-ended adaptation.

- "The issue of open-ended evolution can be summed up by asking under what conditions will an evolutionary system continue to produce novel forms." [391]
- "One perspective on the conditions for open-ended evolution or open-ended learning is thus to consider a rich enough environment which provides an open-ended sequence of problem" [392]

Complex spatiotemporal dynamics. As seen in chapter 2, environments with inherent spatiotemporal dynamics can be a source of complexity through the interaction with adapting agents. Environments with richer and more varied dynamics provide greater opportunities for agents to develop complex adaptations. On Earth, for example, both the Cambrian explosion and the emergence of agriculture have been hypothesized to be driven by sudden environmental changes [325, 393]. While we introduced spatial variations of resource generation in our eco-evolutionary contribution (Sec.2.1, with a gradient of resource generation rate along an axis), enriching the environment with temporal variations could provide richer ground for complex

adaptations and dynamics. This could imply seasonal cycles as in our contribution on agriculture (Sec.2.2), inducing e.g. complex plant life cycles and favoring the discovery of resource conservation strategies to maintain a stock during the "winter season". Such spatiotemporal dynamics can operate at multiple scales, e.g. smaller ones such as the alternation between days and nights (e.g. resulting in different adaptations in diurnal vs. nocturnal animals) or larger ones such as climate changes (which caused massive mass extinctions on Earth [394] and potentially drove early human evolution [394]).

Compositional dynamics. In Sec.3.1, we introduced compositional dynamics in our environments : the possibility to combine elements to produce new ones (as displayed in fig.5.4). Compositional dynamics provide opportunities for discovery during an agent's lifetime, as demonstrated in our contribution 3.1 and other works [184, 251, 390, 395]. Introducing such dynamics into eco-evolutionary simulations like those in Sec.2.1 could enable richer forms of exploration and knowledge accumulation, particularly when agents can learn from each other's discoveries (see Sec.5.2.3 for more details on its potential for cultural evolution to emerge).

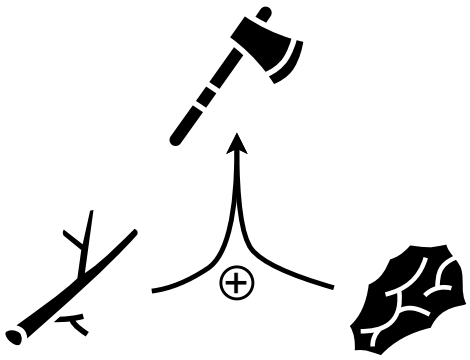


Figure 5.4: **Compositional dynamics**:combining elements to produce new ones. Simple compositional dynamic of the form A+B→C : combining a rock and a stick to build an axe.

Compositional dynamics are even more interesting when they allow building useful tools or functional machines. In this case, agents might have interest to evolve or learn complex exploration strategies, i.e. in the form of intrinsic motivations, as a way to favor relevant discoveries (See Sec.3.1 and Sec.4.4). Such discoveries on how to combine environmental elements to create new ones might provide important benefits, e.g. for niche construction. Ultimately, this drive to explore and try to build functional objects could potentially favor some sort of complex reasoning on how to compose objects. The compositional dynamics could be pre-encoded in the simulation as a list of recipes ($A + B \rightarrow C$, as e.g. in [232, 396, 397]) but could also come from the physics of the system or self-organization dynamics. For example, an accurate 3D physical world could allow complex emergent compositional dynamics, e.g. allowing to make a weapon by sticking a stone in a stick with rope.

Open-ended environments. Ideally, compositional dynamics would be open-ended enabling a never-ending complexification and diversification of technological tools being built. This would therefore require a dynamic that is different from just hardcoding a list of recipes, a limitation of most current works [232, 396, 397].

In the context of environments that enable the open-ended complexification of "technology", the recent work JaxLife [50] introduced programmable robots in a gridworld environment with resource growth dynamics resembling the environments used in our contribution Sec.2. In this system, the agents can learn to program the robots to perform potentially useful tasks. The programmable robots are Turing complete, meaning that they are capable of universal computation. This high expressivity provides a rich ground for more and more complex programs to be learned. In particular, the paper is interested in how high-level open-ended culture and technologies can be evolved in

such an environment. Though the programmable robots are quite abstract and high-level, the rest of the environment dynamics, agents' reproduction, and evolution share similar principles as in our contribution in Sec.2. In particular, they use long non-episodic simulations, potentially allowing complex eco-evolutionary dynamics. They notably highlight strong niche construction dynamics.

Recently some works advocated for the need for *Darwin complete* simulators: *"environmental encoding that can create any possible learning environment"* [3]. Such simulators would enable an open-ended task space, albeit requiring a potentially costly search within this encoded space to find interesting environments.

Towards this goal of a general environment simulator, several works proposed to use neural network based world models[1] [398, 399]. By being guided by language, and trained to generate a huge diversity of environments from which elements might be recombined, these models could allow generating tasks in an open-ended way. Especially, these models are often trained on real world physical environments and therefore could also provide environments with real physics (but are not limited to) which might be useful for some applications such as robot control. However, current models are limited to short-term dynamics, few interactions, and single-agent scenarios, precluding the complex reciprocal interactions between environments and agents discussed in this thesis. They might, however, still be interesting toward the training of generalist agents and will surely improve over the years.

1: World models takes as input a state (often an image) **and an action** to output the next state. $f : S \times A \rightarrow S$

Other works use large language models to open-endedly generate the code or assets configurations of environments in generic physical simulators [46, 65]. Though their open-endedness might be limited by the physical simulator capabilities, further works might use large language models to code any environment from scratch in a programming language.

Complex self-organizing systems as environment dynamics. Another potential candidate for environments that have the potential for open-ended dynamics are self-organizing complex systems such as cellular automata. In particular, some CA like the game of life have been shown to be Turing complete [400], hinting toward their potential for simulating any dynamic.

In addition, cellular automata or reaction diffusion can easily introduce local compositional dynamics. This can be implemented through hardcoded rules converting a mix of elements/chemicals into others, which is classic in reaction diffusion. But it can also be an emergent phenomenon of such complex systems where self-organization lead to the composition of structures, e.g. assembling two patterns together that results in a more complex emergent macro structure. Recent work has shown the capabilities of cellular automata in parameterizing a large space of environments [401], a promising candidate towards a potentially general environment design framework. Our contribution in chapter 1 and especially Sec.1.3 with "multi-species"

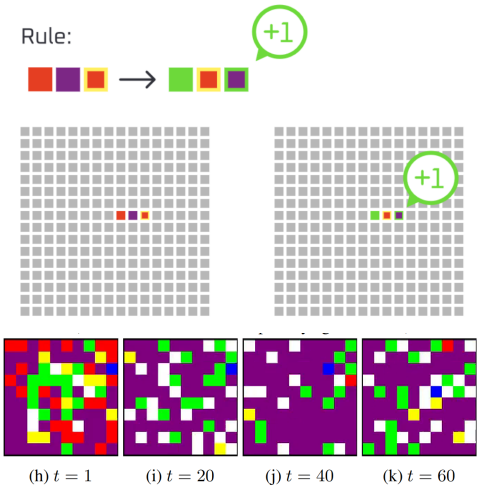


Figure 5.5: Autoverse: cellular automata based environment. Top: example of rule. Bottom: example of dynamic. Figure from [401].

simulations could provide rich parametrized environments with complex dynamics displaying a variety of interacting environmental elements, each with their own properties and potentially adapting along the simulation.

Complex systems like cellular automata are very promising to induce eco-evolutionary feedbacks as small changes can have a huge (potentially structured) impact on the whole environment. This allows the agents to meaningfully impact the dynamics of their environment, potentially resulting in high empowerment. Recently, [402] studied how a population of RL agents, situated in a CA simulated forest fire dynamics, can collectively learn how to harvest trees in order to minimize fire propagation while maximizing the number of trees to collect resources. The environmental dynamics in our eco-evolutionary simulations (Sec.2.1) and emergent agriculture (Sec.2.2) works are in fact cellular automata (and to some extent, any grid-world environment based on local Markovian interactions between neighboring cells). However, this complex system dynamic can also lead to instability in the environment, potentially favoring resource or population collapse.

The fact that self-organizing local rules are often simple, yet lead to great complexity, is also interesting for agents meta-learning to adapt in context. In fact, by being exposed to a variety of rules, the agents might meta-learn to infer the rule by interacting with the environment (as observed in [184]). Agents that can infer environmental rules may gain advantages by exploiting these patterns. However, due to the complex dynamics of such systems, predicting the long-term dynamics from the rule remains challenging. Another way for the agents to adapt in context is to memorize useful patterns discovered during their lifetime (without inferring the underlying rule), potentially building an open-ended repertoire of useful patterns. In addition, discovering interesting patterns and how to make them might necessitate effective exploration mechanisms in the agent.

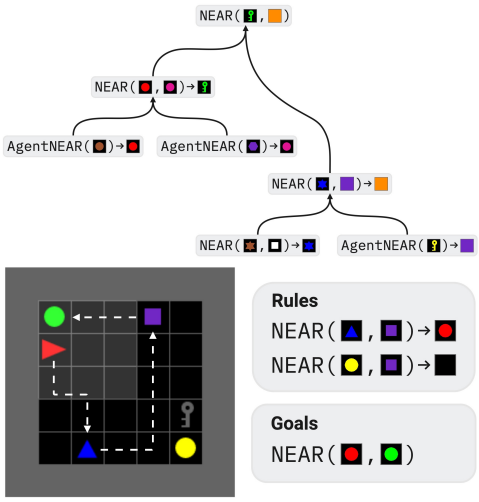


Figure 5.6: Xland-minigrid procedural generation of tasks, including compositional dynamics. Figure from [390].

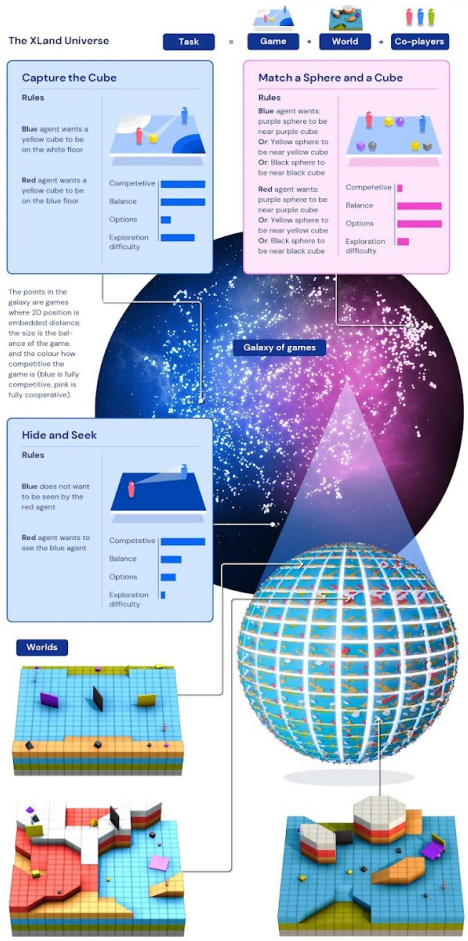


Figure 5.7: Xland procedural generation of tasks and environments. Fig from [44].

Emergent dynamics vs procedural generation of a diversity of tasks. In this thesis, we explored the variability of the environment under two different approaches. The first one is through emergent dynamics as in Sec.1.3 and Sec.2.1: the environmental variation comes from the environment "evolving" through its dynamics and the action of the agents, in particular through long non-episodic simulation in rich environments. The other approach, used in Sec.1.2 and Chap.3, relies on procedurally generated episodic (often short-term) tasks from a parametrized simulator.

Procedural generation of tasks is mainly used in machine learning to train generalist agents [44, 45, 390, 403]. This method is easier to control; the experimenters can easily set the task as they intend and prespecify variations. This enables researchers to test specific mechanisms while exposing agents to controlled tasks. However, this in turn potentially limits the space of tasks to the space that was "imagined" by the experimenter and thus can be limiting and not totally open-ended.

Recent works with Darwin complete simulators (or at least very general simulators), allow procedural generation of a very large repertoire of environments. However, the control over the training tasks is then limited and also necessitates exploring a very large space to find interesting environments. In particular, works often use unsupervised environment design (UED) [195, 196, 199] techniques to explore this vast space and find relevant training environments for the current capabilities of the agents.

On the other hand, artificial life works sometimes use emergent dynamics as the driver of variability in the environment, using a single environment whose dynamics generate all the variations [50, 201]. Having a single environment for a long time allows for long-term dynamics to emerge, such as eco-evolutionary dynamics and culture accumulation. This also allows for environments that would have been hard to design, which rather emerge from the discoveries made by the agents themselves. With rich enough environmental dynamics, the environment can lead to open-ended dynamics by allowing never-ending novel states and opportunities.

However, the initial richness of the environment dynamic might be as well limiting. In addition, as the environment is not reset, the environment is therefore more unstable and can lead to collapse especially as incremental change can accumulate. Also, adapting agents are known for their abilities to exploit flaws in the design of environments (especially in RL) [404], which is hard to predict when designing a single complex environment.

Techniques used in the search for parameters in procedural generation of environments such as unsupervised environment design or quality diversity could also be applied to find environments supporting long-term rich dynamics. Exploring the space of environments in a Darwin complete simulator could in fact be guided by certain metrics of the entire long simulation dynamics such as evolutionary activity, complexification of the environment and/or behavior of the agents. However, finding relevant metrics of such quantities reveals to be challenging, especially in environments with emergent complexity that was not predicted and is constantly changing (see Sec.5.3.2 for more details on the challenges of measuring complexity and open-endedness). Exploring in this manner, however, requires running long simulations for each environment dynamic tested, leading to a potentially huge computational cost.

As done throughout this thesis, we also believe that isolating certain mechanisms to more profoundly understand the environmental factors favoring the emergence of specific behaviors is important in comprehending the building blocks necessary (or useful) for creating large-scale complex environments that would lead to open-ended dynamics. In particular, toward the emergence of generalist agents, further work should be done on trying to understand the conditions for general adaptation and in particular general curiosity to emerge. For example, we started to elicit in preliminary work Sec.4.4, simulations displaying first hints of the emergence of information-seeking behavior as a way to deal with variability during the episode. Finding environments leading to the emergence of general reasoning is also

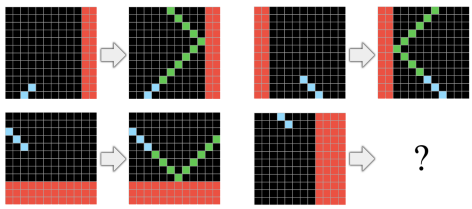


Figure 5.8: Arc reasoning benchmark example. The agent is exposed to a handful of transformation examples (input and corresponding outputs, shown on the left and right sides of the arrows, respectively). Informed by these examples, the task is to infer the output (question mark) of a novel input. This requires strong few-shot generalization abilities, similar to IQ tests. [405]

an interesting direction, for example with RL environments inspired by the ARC benchmark [405, 406] (Fig.5.8).

We refer to section 5.3.1 for additional insights on the balance between engineered bias and emergent complexity.

Curriculum and serendipity. Ideally, environments should facilitate a progressive curriculum of adaptation or discovery. However "intelligent" it can be, any agent will have trouble to adapt if the gaps in difficulty between successive environmental challenges are too large (this is the case for humans too). For example, if discovering a new tool requires precisely combining multiple rare resources in a specific sequence, agents may never bridge this gap through random exploration alone. Instead, environments should offer intermediate stepping stones of increasing complexity. Ideally, environments should foster serendipity, enabling agents to randomly discover mechanisms that they can subsequently learn to exploit. This is especially true as current adaptation mechanisms lack proper efficient exploration mechanisms, for example in reinforcement learning or evolutionary algorithms where exploration relies on random low-level actions or mutations. However, advances in intrinsically motivated agents [134] and emergent meta-learned exploration (Sec.3.1, [44, 184]) could provide useful exploration mechanisms to overcome this limitation to some extent. In particular, intrinsically motivated agents can produce their own curriculum, for example through exploration strategies maximizing learning progress [407, 408].

Scaling environments. Finally, recent advances in both hardware and software have enabled the expansion of environmental simulations and training processes to unprecedented scales. In particular, most of the works presented in this thesis used the JAX framework [289] allowing for the simulation of large-scale grids with a huge number of agents. Parallelizing environments on GPU enabled huge speedup in reinforcement learning and evolutionary algorithms [103, 174, 290]. This enables the simulation of complex multi-scale adaptations on a diversity of environments, such as combining RL and evolutionary algorithms (as e.g. in our additional contribution Sec.4.2). Future research should prioritize the development of efficient, large-scale environments. These environments could enable reinforcement learning and evolutionary algorithms to leverage massive parallelization and computational resources, following the successful scaling patterns seen in supervised learning.

5.2.2 Agent design

Throughout this thesis, we explored how artificial agents can adapt to varying environmental constraints and opportunities at different scales. Chap.1 adopted a radical enactivist view focusing on low-level adaptation at proto-morpho-cognitive (Sec.1.2) and proto-evolutionary (Sec.1.3) scales. Chap.2 and 3 switched to a more standard mechanistic view focusing on intergenerational evolutionary (Sec.2.1) and

individual-life developmental (Sec.3.1 and Sec.3.2) scales. In this section, we discuss limitations and perspectives related to these agent design choices.

Fixed architecture. The works we presented in chapters 2 and 3 use a fixed neural network architecture which can potentially limit their opportunities to increase in complexity. Research in neuroevolution has explored evolving both neural architectures and their connection weights simultaneously [101, 409]. This allows networks not only to optimize their parameters, but also to structurally adapt to new challenges, providing greater flexibility than weight evolution alone. Building on this idea, some researchers have developed meta-learning approaches that learn the rules for generating and growing neural network structures – inspired by the biological development of the brain – known as neural developmental programs [56, 410]. Notably, they also propose a framework in which the architecture evolves and self-organizes dynamically during the lifetime of the agents, depending on their interactions with the environment. This approach, termed lifelong neural developmental programs [57], emphasizes continual learning and structural adaptation. Our additional work Sec.4.2,follows the idea of evolving the architecture and meta-learns a minimal parametrized encoding of the macro-level properties (such as the amount of connectivity or recurrency) of an echo state neural network [2] [411], which is randomly generated at each instance. The meta-learned parameters are optimized so that every echo state neural network generated with these parameters allows versatile fast adaptation through RL on a diversity of tasks.

2: **Echo state neural networks**(ESSN) [411] are a type of recurrent neural network: "The main idea is (i) to drive a random, large, fixed recurrent neural network with the input signal, thereby inducing in each neuron within this "reservoir" network a nonlinear response signal, and (ii) combine a desired output signal by a trainable linear combination of all of these response signals." [412]

Another recent and promising direction in the adaptive generation of agent's cognitive architectures relies on LLM-based code generation. This approach enables the flexible synthesis of entire controllers and even learning algorithms in programming language space (which is, by essence, an open-ended design space) [413, 414].

Brain-body coevolution. Another limitation of our works in chapters 2 and 3 is the fact that the bodies of the agents are fixed and identical. This might also limit the opportunity for the agents to adapt in different ways. In particular, through the concept of morphological computation [240, 415] – the fact that the body also participates in the cognitive "computation" or reduces the complexity of the control – Pfeifer et al. shed light on the central contribution of the body in cognition (in line with the enactive view). Brain-body coevolution is both a historical and currently active topic in ALife [416–421].

Allowing the bodies of the agents to evolve across generations or develop through individual lifetime could introduce a greater diversity of capabilities and morphologies, enriching the interactions between agents and enabling specialization to distinct ecological niches.

This thesis has, in fact, mostly explored the two extremes of the spectrum. On the one hand, Chap.1 adopted a rather radical enactivist approach, avoiding any pre-defined dichotomy between agents and environment, or between sensors, actuators and brain. On the other

hand, Chap 2 and 3 adopted a more standard mechanistic approach, with predefined and fixed morphologies and cognitive architectures. Mixed approaches, considering fixed morpho-cognitive building blocks being adaptively composed into diverse architectures, have been proposed [75].

Self-organizing systems as a general design space. As shown in chapter 1, self-organizing systems, such as cellular automata, can be an interesting sandbox to evolve body morphologies and cognition as emergent properties of a self-organizing structure. This approach is in line with recent propositions in evolutionary and synthetic biology to study morpho-cognitive systems in all their diversity and generality, which is sometimes referred to as *basal cognition* [239, 264]. In such views, unicellular organisms [422] or even gene regulatory networks [423] can be conceived as cognitive systems able to make decisions and even learn from experience. We believe that the computational methods we have proposed in Chap 1 can provide a useful experimental testbed to simulate and test hypotheses in basal cognition.

In AI and ALife, the methods we developed in Chap 1 may set the ground for a completely different approach to building open-ended and versatile AI systems as compared to current deep learning and generative AI approaches – which still either assume prior notions of agency and embodiment, or completely ignore them. Here we aim to address how to build artificial systems where sensorimotor agency and simple forms of learning and evolution self-organize from scratch.

Meta-evolution and genotype-phenotype mapping. Meta-evolution refers to the process by which evolutionary mechanisms have themselves (macro-)evolved (e.g. the structure of the DNA and how it encodes phenotypic traits). This thesis has mostly ignored such mechanisms, with the exception of Sec.1.3 where we studied the self-organization of proto-evolutionary dynamics. In nature, this includes changes in evolvability [424] and modifications to genotype-phenotype mappings. In particular, complex (constantly evolving) genotype-phenotype mapping might allow effective mutations that generate very diverse and efficient phenotypic variations. Several works in computer simulation explored the questions of the evolution of evolvability [302] and genotype-phenotype mapping [425–427].

5.2.3 Going further in the complex interactions between groups of agents

This thesis did not explore cultural evolution as an adaptation mechanism, except to some extent in the contribution in Sec.3.2, which addresses cultural aspects through the emergence of a common lexicon and shared intentionality. In this section, we explore how cultural evolution (and especially the cultural accumulation of knowledge) could emerge in a simulation.

Culture. Building on our non-episodic eco-evolutionary framework (Sec. 2.1), we investigated the emergence of altruistic behaviors directed toward offsprings (Sec. 4.1). Future studies could extend this work by examining the formation of larger social groups beyond the immediate parent-offspring relationships. In addition, the observed altruistic behavior in our study was limited to resource sharing—a predefined, hardcoded action that agents evolved to perform. Expanding on this, future research could explore a broader range of altruistic behaviors, such as the transmission of information, which may reveal additional mechanisms driving the evolution of cooperation.

A particularly intriguing research direction is to investigate how cultural transmission and knowledge accumulation might emerge spontaneously in artificial systems. While recent work has demonstrated that agents can meta-learn to acquire knowledge by observing their peers [150], these studies were conducted under highly controlled conditions: agents were trained on episodic training within predefined tasks and utilized oracle policies to meta-train the learners. A key open question is whether similar cultural learning processes could emerge naturally in a less engineered eco-evolutionary context (Sec.2.1), where agents must simultaneously survive, adapt, and potentially develop the capability to learn from each other without external guidance or supervision.

In particular, the compositional dynamics we introduced in the previous section Sec.5.2.1 could be an interesting ingredient favoring the emergence of cultural transmission. For example, recipes that are hard to discover in the lifetime of the agents but have beneficial impacts should benefit from being culturally transmitted across generations. The emergence of efficient (potentially open-ended) cultural accumulation in groups of agents in an embodied eco-evolutionary simulation would be interesting both as a result and as a mean to study further the dynamics of cultural transmission (with movements of populations etc.).

Further studies could also explore the emergence of active teaching, in which agents deliberately perform actions with the goal of transmitting information.

Communication. Cultural transmission and especially teaching might be facilitated by adding explicit communication channels in our simulations, such as continuous signals that agents can produce and perceive. Although agents in our current simulations can in theory use movements and environmental changes to transmit information (i.e. implicit communication), introducing richer communication mechanisms could facilitate more structured interactions and enable the emergence of more complex cooperative behaviors. Communication could also be used to form groups and coordinate behavior, as presented in our work Sec.3.2. Further works could study the impact of richer communication on the emergent social behavior observed in eco-evolutionary simulations such as the ones presented in section Sec.2.1.

Having developed our perspectives on the specific roles of environment, agent and interaction design in the quest of open-ended dy-

namics in artificial systems, we will now conclude this thesis on general and longer-term – perhaps also more speculative – perspectives.

5.3 General perspectives

We presented in the previous section concrete perspectives on the architectural components we believe are interesting in the quest for open-endedness. We will now discuss general perspectives about emergent complexity and open-endedness.

5.3.1 Balancing emergent complexity and engineered dynamics

Ideally, to introduce as few engineered human biases as possible and observe the whole open-ended picture unfold from basic chemical reactions as it did in the real world, we would like to have everything emerge from a basic physical simulation, for example similar to the direction taken in chapter 1.

The example of Flow Lenia Sec.1.3. However, for now, the emerging dynamics in chapter 1 are still far from leading to the high level of cognition we observe in the real world or even in classical machine learning work. Further work could explore how learning and memory could emerge in such a cellular automaton system, potentially taking inspiration from meta-learning experiments in classical machine learning (like the one performed in Sec.3.1).

While our work in Sec.1.3 shows promising results in emerging evolution, knowing if the system already contains the sufficient conditions to enable truly open-ended evolution is still an open question. Without surprise, we suspect it does not: we ran very long-term simulations and so far, they all converge to a plateau in diversity. Yet, it is still a possibility that more scaling (in terms of size of the grid or even longer timescale) and different initial parameters could unlock a richer evolutionary dynamics. Further experiments, using e.g. quality diversity to explore the space of parameters, and larger-scale experiments need to be conducted to explore this direction.

Another possibility is that the current physics of the system does not allow for open-ended dynamics, for example, as it may systematically lead to the existence of a single "best" species that just overtakes every other one without having any "counter species". Having a system which always guarantees the possibility of having a counter should, for example, prevent the winner-takes-all result that we see in long-term experiments, and could potentially lead to a pressure to complexify or diversify (Sec.0.3.3). In particular, biologists have shed light on the role of parasitism as a driver of continual change and potential increase in complexity [216]. A similar dynamic has been reproduced in silico: in Alife simulation of artificial chemistry [428], as well as minimal computational machine ecosystems [429].

The system may also require the introduction of additional low-level physical elements/rules, or higher-level environmental dynamics. For instance, creating distinct high-level ecological niches with varying demands could encourage the evolution of diverse capabilities. The choice of the elements to be introduced could be inspired by the more controlled eco-evolutionary experiments done in Sec.2.1. Introducing a small number of engineered environmental elements could help identify missing components in our simulations. This approach may reveal whether these components can emerge naturally or indicate the need for a more general system.

General thoughts. Throughout this thesis, we have argued that introducing engineered high-level structures can provide valuable insights into the mechanisms and ingredients required for open-ended dynamics. These insights, in turn, may guide the design of low-level systems capable of generating complexity from the bottom up.

For example, in recent works, a lot of effort has been put on using large language models to introduce high-level components, such as task generation [46, 48, 65] or diversity search [430], that aim for open-ended dynamics. These experiments with high-level components may allow us to elicit certain mechanisms and potentially understand better the mechanisms of open-endedness.

On the other hand, the emergent complexity approach can lead to more efficient systems by enabling the discovery of dynamics or solutions that are difficult to engineer or anticipate [3, 10]. The emergent solutions may also lead to the transfer of ideas to high-level engineered systems[3]. In particular, understanding the conditions that drive complexity in simpler systems could provide valuable insights for designing more complex, high-level engineered systems.

3: For example Lu et al in their paper "Discovered Policy Optimisation" [174] used meta-learning to discover efficient reinforcement learning algorithms, ultimately using the discovered insights to also formulate a new reinforcement learning algorithms

In fact, emergent complexity and engineered biases benefit each other and are often complementary, with engineered elements often forming the basis for complexity to emerge. It is for instance clearly the case in [44](illustrated Fig.5.7) where most of the environmental dynamics is carefully engineered, yet enables the procedural generation of a wide diversity of tasks promoting the learning of general and complex skills. However, the engineered high-level structure might often limit the space of possibilities by restricting it to a space that was envisioned by the experimenter. This can therefore potentially limit the emergence of complexity.

We also observe more and more open-ended task generation used to train high-level cognitive architecture for the learning of general skills [44–49]. In fact, due to the limited amount of human-generated data, machine learning methods are increasingly moving beyond supervised training by incorporating open-ended data generation [3, 43, 47]. The question on how to generate such data is therefore central.

5.3.2 The challenges of measuring and analyzing open endedness

As mentioned in the introduction, the notion of open-endedness does not rely on a final objective but rather on emergent complexity, also often meaning that we might not have anticipated the process that might emerge. This raises the questions : how to know when a process is open-ended ? Is there a measure of it ? In fact, as we don't know what we're looking for exactly, in terms of dynamics, defining a measure of open-endedness proves to be challenging.

Several attempts to formalize a measure of open-endedness have been proposed, including Bedau et al. who proposed a measure focused on evolutionary dynamics and based on evolutionary activities statistics (similar to what we used in Sec.1.3) [431]. However, while some simulations have passed this test of open-endedness at the highest level [35, 36, 432], they fall short of matching the open-endedness observed in natural or cultural evolution. Even if Bedau's test already emphasizes that there are different levels of open-endedness (i.e. open-endedness is not a binary property), the fact that such simulations still pass the maximum level highlights the need for more complete measures of open-endedness. In addition, Bedau's test focuses on evolutionary dynamics and is therefore nontrivial to apply to other dynamics. Measuring open-endedness is in fact still an open question with several challenges [433].

The concept of open-endedness is closely tied to novelty [392], which is inherently abstract and subjective. Different definitions of novelty can lead to different interpretations of what constitutes open-endedness. For example, the noisy TV example – a screen that displays every few instants a new sampled image where each pixel follows a normal distribution – might be considered novel under a specific measure of novelty (as the probability to sample an image that has already been sampled is 0), but from a human-centered point of view, this would not be considered as really open-ended.

Similarly, the sequence of natural numbers – 1,2,3... – always brings new elements but would not be considered open-ended by a human. In fact, natural numbers can be summarized into a single concept where additional integers are not really considered new. With the same idea, the noisy TV can be modeled by a statistical model. To address this issue, Hughes et al. define as novel something that can't be predicted by a statistical model learned on the discoveries so far : *"[...]what an observer considers 'new' should be artifacts that are unpredictable according to their current statistical model of the system under consideration. Moreover, we specify that the observer's 'perspective' is generated by learning that statistical model on the history of artifacts thus far presented by the system."* [43].

With a similar idea, Stepney and Hickinbotham state that detecting open-endedness in a system is in fact an open-ended process on its own [434], as :"an open-ended system will eventually move outside its current model of behavior, and hence outside any measure based on that model" [434].

Building on that idea of changing novelty, works in machine learning and diversity search have proposed to learn the diversity space in an online manner, adapting at each new discovery. For example, a possibility is to learn a neural network based encoding of the stream of discoveries with an autoencoder which will be used as the behavioral space to explore [234]. In particular, Etcheverry et al. also proposed to iteratively build a diversity of diversity (meta-diversity, Fig.5.9), creating a new space of behavior when the previous ones were saturated hence not capturing well enough the new behaviors [235] – in line with the idea of having always "new novelty". Those approaches revealed to be effective at exploring more effectively.

As emergent phenomena are usually unknown a priori, we often relied on the design of a-posteriori measures based on our own observations of the simulations. We most of the time used qualitative observation or general measures to track emergent complexity and then designed more specific measures to further characterize the dynamics. This approach is relevant as most of our contributions often focused on isolating specific transitions and emergent complexity, but its generality is quite limited. Designing methods for the generic detection of phase transitions in complex systems is still an open problem.

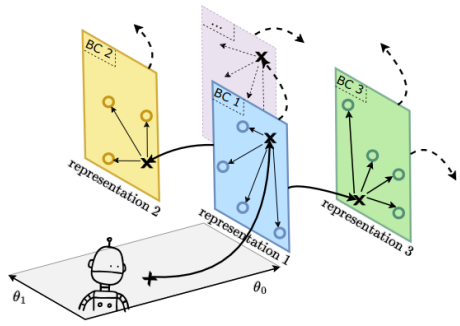


Figure 5.9: **Meta diversity**. An agent iteratively learns an ever-growing diversity of representations to explore. Figure from [2].

Finally, though they might provide useful insights into a system, measures of open-endedness are inherently limited to testing novelty over finite timeframes. As a result, they offer little information about a system's potential for infinite novelty generation. They might, however, still be useful as a proxy to search for dynamics that seem to indicate open-ended dynamics or at least some level of emergent complexity.

5.3.3 Simulations to understand the real world

While this thesis primarily explores simulation in the context of open-ended dynamics and the development of generalist agents, the simulated dynamics studied here could also contribute to a deeper understanding of real-world phenomena. Simulations provide a powerful tool for hypothesis testing, allowing researchers to explore variations and assess potential outcomes.

In particular, the research presented in Chapter 1 offers insights into how groups of cells can self-organize through local interactions, especially during morphogenesis—the process of body development. These contributions may enhance our understanding of low-level "cognition" in cellular systems and how cells make collective decisions at a macroscopic scale. For instance, this perspective could provide new insights into cancer, a condition in which a subset of cells escapes the collective order to establish its own independent individuality. Understanding and reverse-engineering the principles of cellular self-organization could lead to significant advancements in medicine [268, 269]. Finally, the findings in Chapter 1 may also inform broader discussions on the emergence of individuality and evolutionary dynamics, shedding light on how life could arise from an initially lifeless system governed solely by physical laws.

Similarly, the evo-evolutionary simulations we proposed in chapter 2 can contribute to a better understanding of niche construction phenomena, as well as to test hypotheses in behavioral ecology. In particular, our work on simulating the emergence of agricultural practices in agent-based systems (Sec.2.2) was designed with this goal in mind. Further work should be done to connect more deeply our contributions in artificial life and machine learning to other fields such as biology, (human) behavioral ecology, or neuroscience.

More generally, at a time of environmental crisis induced by human activity, a better understanding of the reciprocal causation between environmental dynamics and agent's adaptation is an important research direction. Artificial ecosystem simulations can potentially help to explore the vast space of potential scenarios and provide a deeper understanding of the problem in all its generality (i.e. not limited to human-induced global warming and biodiversity loss).

Appendix

A

Appendix

A.1 Appendix: Discovering sensorimotor agency in cellular automata

- In the first part of this appendix, we provide several additional results :
  - In section A.1.2, we provide the resulting curriculum "phylogeny" from a run of IMGEP.
  - In section A.1.3, we provide ablation of the IMGEP method: removing obstacles from the training in A.1.3, replacing the gradient with a simple evolutionary algorithm in A.1.3, and replacing the biased goal sampling by an uniform goal sampling in A.1.3.
  - In section A.1.4, we provide results for each of the 10 seeds to display the variability.
  - In section A.1.5, we provide the full results for the generalization tests.
- We then provide the details of the method, system, and tests :
  - In section A.1.6, we describe the Lenia system in details. In particular in subsection A.1.6, we describe the change made on the original lenia system from [8, 9] to make it more differentiable.
  - In section A.1.7, we describe the IMGEP method in details.
  - In section A.1.8, we provide details about the tests and measures used in the main papers: empirical agency test in A.1.8, moving test in A.1.8, speed measure in A.1.8, basic obstacle test in A.1.8, generalization tests in A.1.8.
  - In section A.1.9, we provide details about the baselines we use for comparison: random search in A.1.9, agent from the original lenia papers [8, 9] in A.1.9
- We provided in section A.1.10 the legends of the movies.

A.1.1 Data availability

The resulting parameters as well as their measured performances on the test tasks are available on Github at `https://github.com/flowersteam/sensorimotor-lenia-search` in the data folder. More precisely:

- Folder *imgep_exploration* contains parameters generated by the IMGEP method presented in the main text as well as their measured robustness.

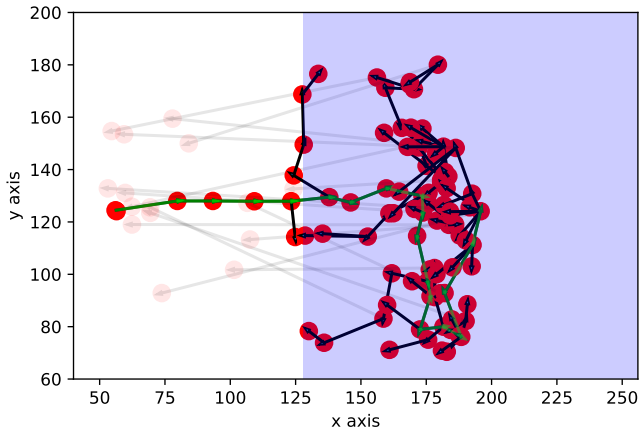


Figure A.1: "Phylogeny tree" of one run of IMGEP. The red dot are reached positions (by a step of IMGEP). The blue zone correspond to the zone where obstacles can be placed. Black arrows indicate optimization progress (the point at the end of the arrow was obtained after optimizing the one at the start of the arrow). The path leading to the best agent (reaching the furthest position on the x axis) is highlighted in green. Interestingly we can see that the best path is not necessarily a straight path. For visibility reasons, we put transparency on the optimization steps that led to reached positions far from the reached position of the parameters that was used to initialize the optimization (often due to failing).

- Folder *random_exploration* contains parameters generated by random exploration as well as their measured robustness.
- Folder *handmade_exploration* contains parameters from the original Lenia papers [8, 9] (more details in appendix A.1.9) as well as their measured robustness.
- Folder *imgep_no_grad_init_exploration* contains parameters obtained from the IMGEP with ablation on the gradient (described in appendix A.1.3) as well as their measured robustness.
- Folder *imgep_no_obstacles_exploration* contains parameters obtained from the IMGEP with ablation of the obstacles (described in appendixA.1.3) as well as their measured robustness.
- Folder *imgep_random_sample_init_exploration* contains parameters obtained from the IMGEP with a uniform sampling of goals (described in appendix A.1.3) as well as their measured robustness.
- Folder *videos* contains all video presented in this work.
- File *creatures_categories.json* contains the result of the agency and moving test for all the pre-filtered parameters (more details on the pre-filter in appendix A.1.8) from the IMGEP, random, handmade exploration and "IMGEP no obstacles".
- File *creatures_categories_ablation.json* contains the result of the agency and moving test for all the pre-filtered parameters (more details on the pre-filter in appendix A.1.8) from the ablations presented in appendix A.1.3 and A.1.3.

We also provide the code to reproduce the experiments on Github at `https://github.com/flowersteam/sensorimotor-lenia-search`.

A.1.2 Curriculum phylogeny

In Fig.A.1, we explore the curriculum path that is generated by the IMGEP. For this aim, we plot the achieved position (reached goal) by each step of the IMGEP. Arrows show, for each step, what was the previous step result used as initialization. In addition, we highlight in green the sequence of reached positions leading to the furthest position attained. We observe that the path to this furthest position is far from being straightforward. This indicates a rather complex optimization landscape toward this position, that would have been difficult to

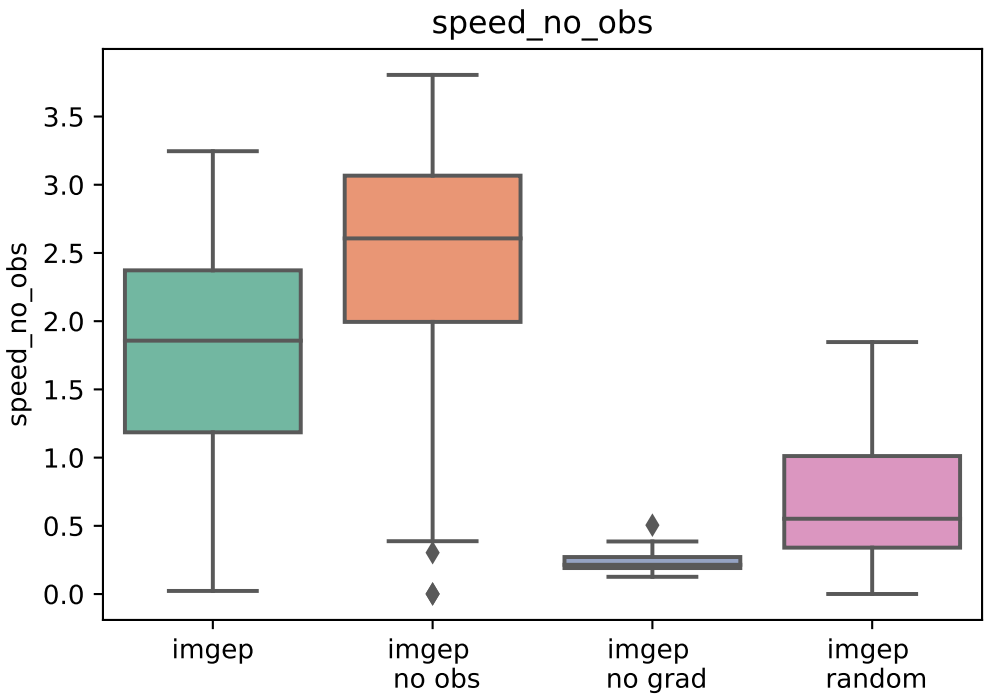


Figure A.2: Comparison of ablation on speed

navigate through gradient descent alone. By generating diverse goals and their associated solutions in parameter space, the IMGEP is able to explore potential stepping stones that can later on prove useful to reach difficult positions.

A.1.3 Ablations

We will call the training procedure described in the main text as the *original method*, to which we provide additional detail in A.1.7. In this section, we provide ablation studies aiming to evaluate the effect of removing different components of this original method. To make it as fair as possible and also highlight the difference each ablation introduces, all ablation studies except the "IMGEP no obstacle" were made starting with the same initialization of the history as the ones obtained from the initialization search (7) of the original method. This initialization might however be influenced by the presence of obstacles, this is why "IMGEP no obstacle" will run its own initialization search.

IMGEP no obstacles

In this ablation, we use the same training procedure as in the original method but remove the obstacles from the grid. This means that during training the agent will only be trained to go further but will never encounter any obstacle.

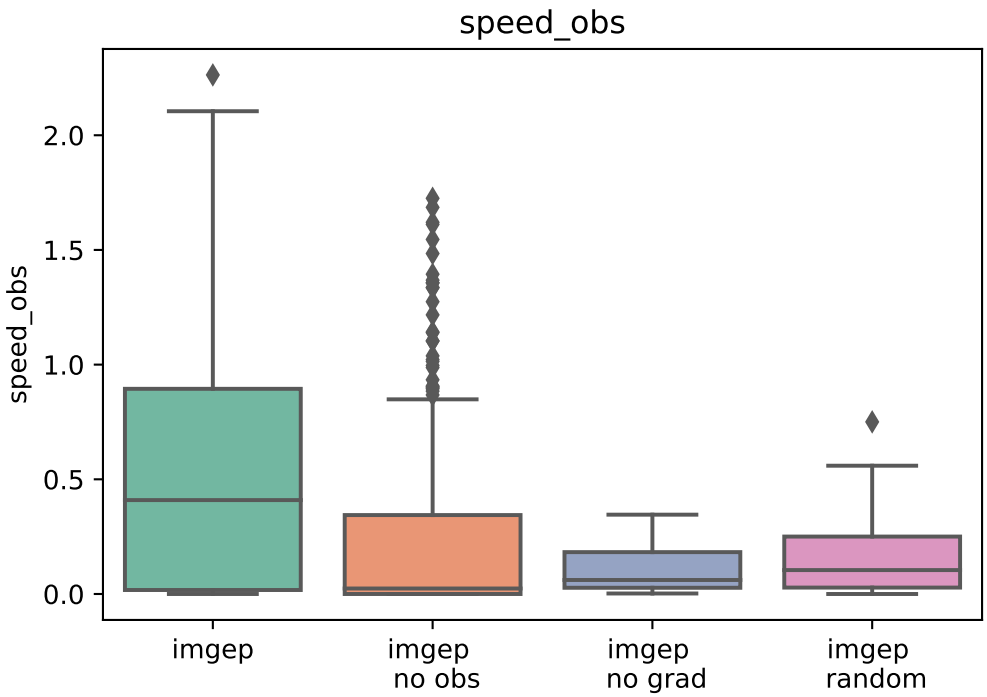


Figure A.3: Comparison of ablation on speed with obstacles

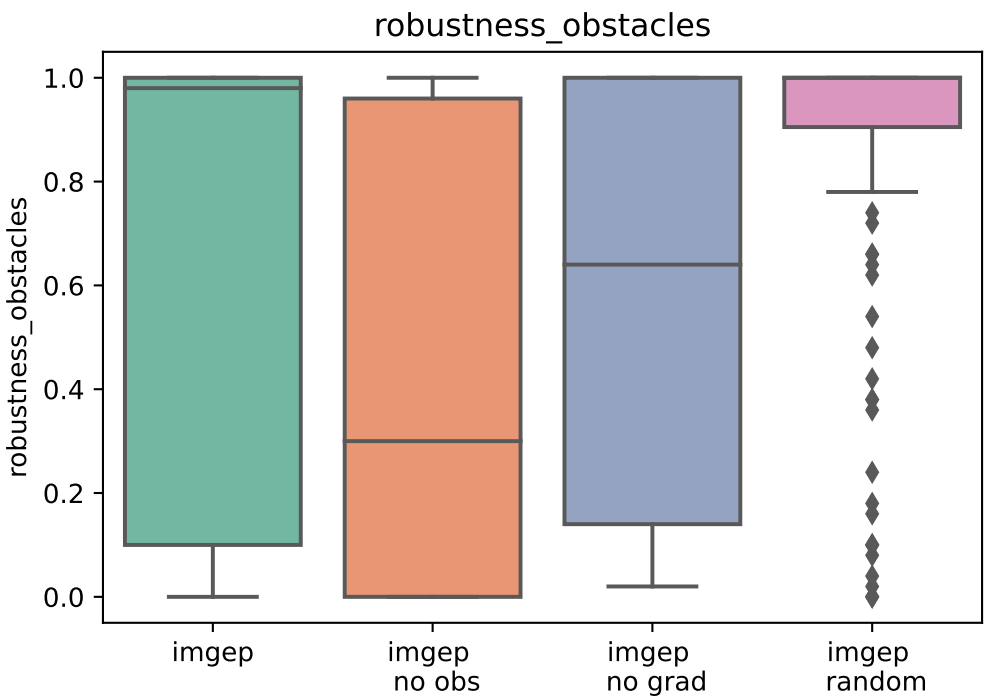


Figure A.4: Comparison of ablation on robustness to static obstacles

With this ablation, we obtain moving agents that are faster without obstacles than the original method (Fig. A.2) but have far less robustness to obstacles (Fig. A.4) and especially here against moving obstacles (Fig. A.5,A.6). We also observe that agents trained in the

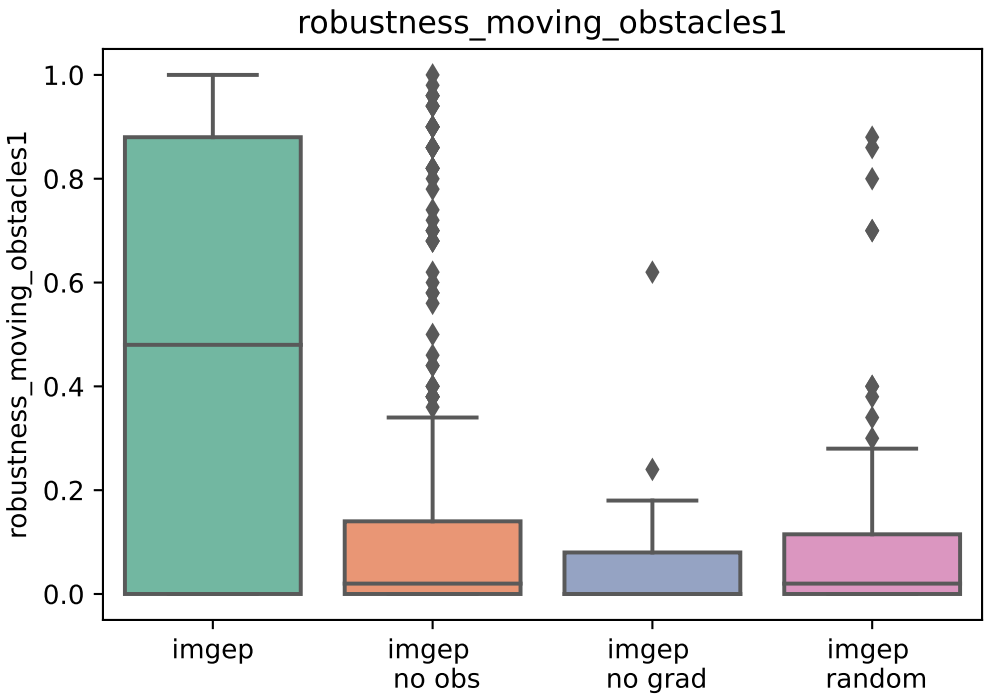


Figure A.5: Comparison of ablation on robustness to moving obstacles of speed 1

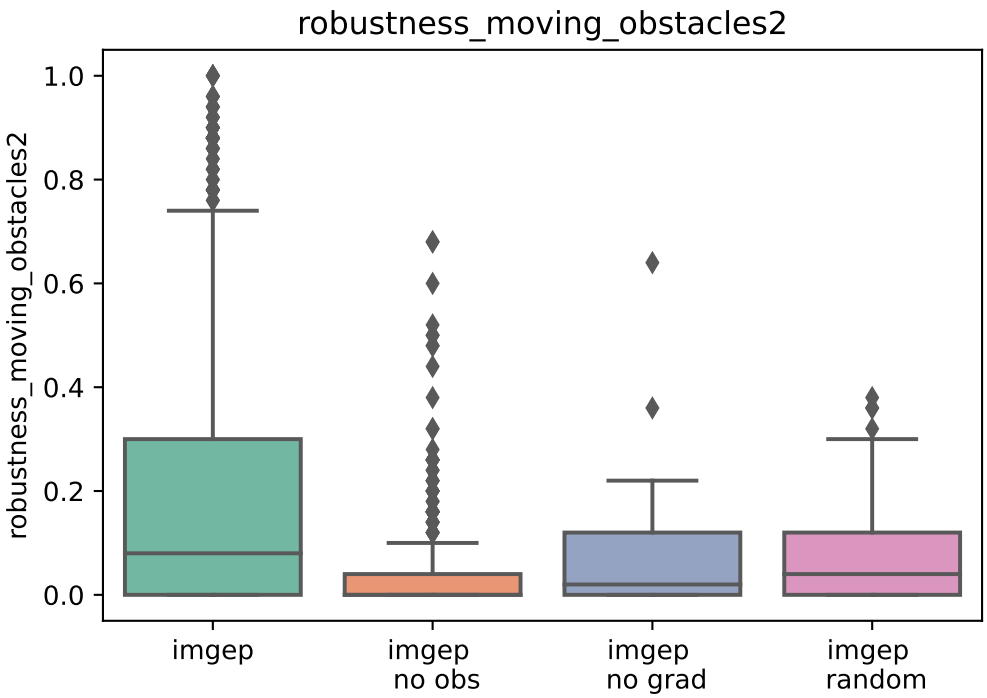


Figure A.6: Comparison of ablation on robustness to moving obstacles of speed 2

original condition, at equal speed, are more robust to obstacles than those in this ablation (Fig.A.7). This is intuitive as the training without obstacles facilitates reaching further positions (as there is no obstacle in the grid), resulting in higher speed since the episode duration

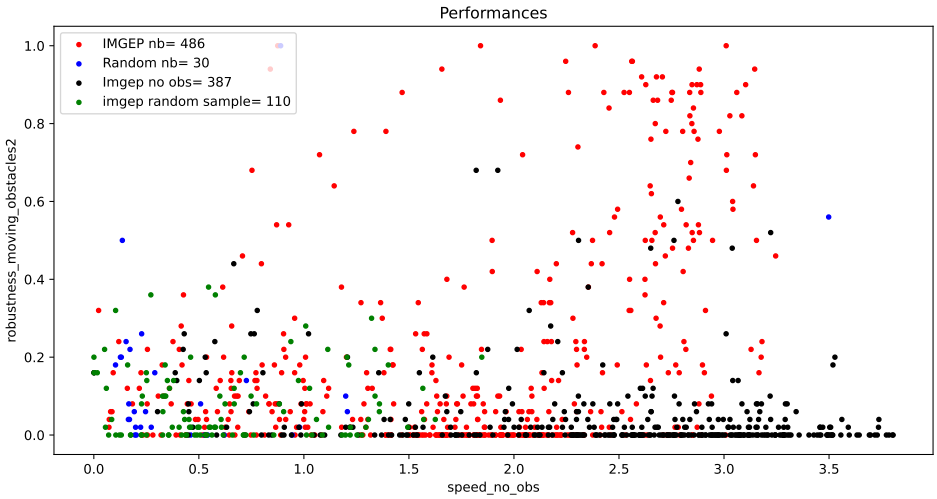


Figure A.7: Scatter plot of robustness to moving obstacles of speed 2 (y) and speed without obstacles (x) of IMGEP (red), IMGEP without obstacles in the search (black), Random search (blue) and IMGEP with random sampling of goals (green). Even for moving agents with comparable speed without obstacles, IMGEP with no obstacles has far less robustness to obstacles of speed 2 than IMGEP trained with obstacles.

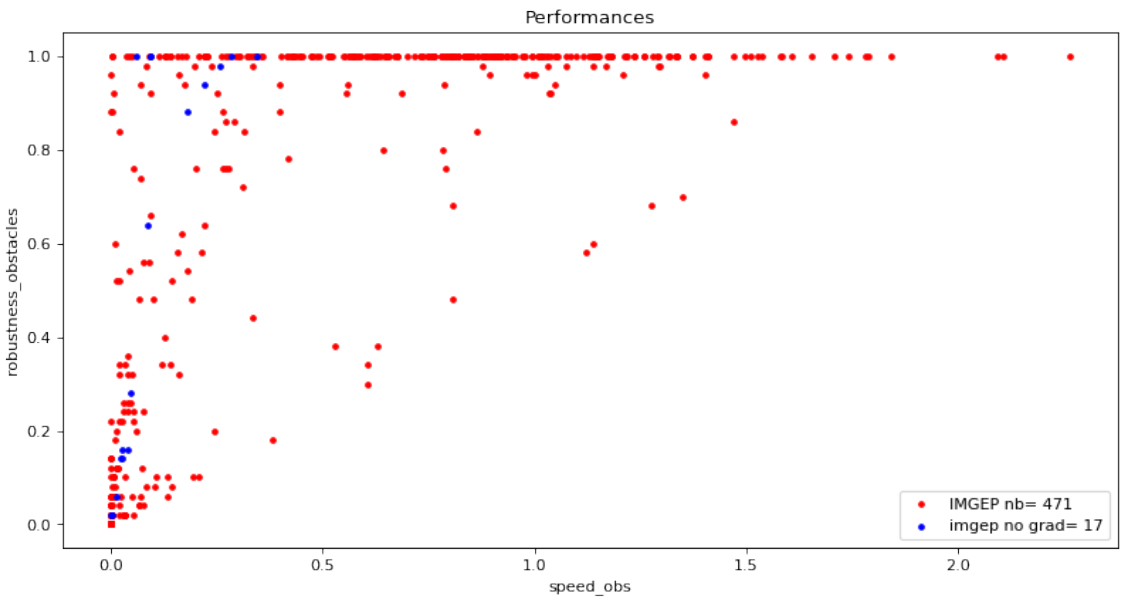


Figure A.8: Comparison of the original IMGEP method and an IMGEP where gradient descent optimization of the parameters is replaced by random mutations as described in A.1.3. We can see that random mutation hardly succeed in optimizing the parameters leading to very poor performance compared to the IMGEP with gradient descent.

remains constant. However as they are not optimized to resist obstacles, we observe much lower robustness.

No gradient

In this experiment, we replace the gradient descent in the original method by a simple evolutionary strategy. For each goal we replace the gradient descent by several parallels trials of random mutation (mutation as described in 11) from the candidate parameters with a

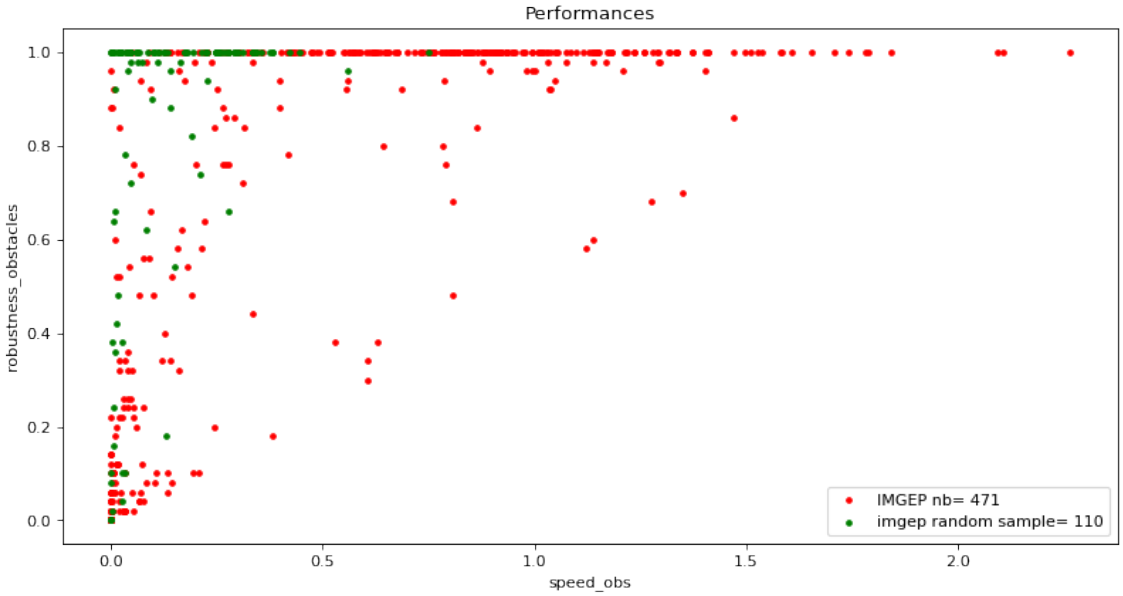


Figure A.9: Comparison of the original IMGEP method and an IMGEP where our our biased goal sampling is replaced by a random sampling of goal in the grid as described in A.1.3. We can see that our biased sampling is much more efficient at finding robust fast moving agents.

number of trials equal to the number of gradient descent steps performed during optimization in the original method. At the end of those trials we select the parameters having the lowest loss regarding the goal (same loss as the one used for gradient descent in the original method). We observe that the performances of this method is significantly lower (Fig.A.8), suggesting that random mutations is not effective in such hard optimization landscapes (and especially with such little number of rollouts) and leads in most cases to explosion or vanish of the matter.

Uniform Random sampling of target in IMGEP

In this experiment, we replace the curriculum-driven goal sampling of the original method (detail on curriculum in 7) by a uniform sampling in the grid.

Compared to the original method, we observe overall lower performances in term of speed and robustness (Fig. A.9 and Fig.A.2,A.3,A.5,A.6). This can be explained by the fact that random sampling often sample goals that are impossible to reach at the time. We observe that, with the same budget as the original method run, it only reaches a very small subset of the entire grid compared to the original method (Fig.A.10). Most target goals far from initialization failed while goals that were close enough were sometimes successful.

However, we observe that this ablation still allows to obtain more moving agents than random search (110 vs. 30)

We introduced a curriculum in our original method mostly to speed up computation. We indeed show with this ablation how it benefits the search process. Note however that, in theory, the current ablation should obtain similar results if given enough compute budget (but will most likely require much more time). In fact, a curriculum can

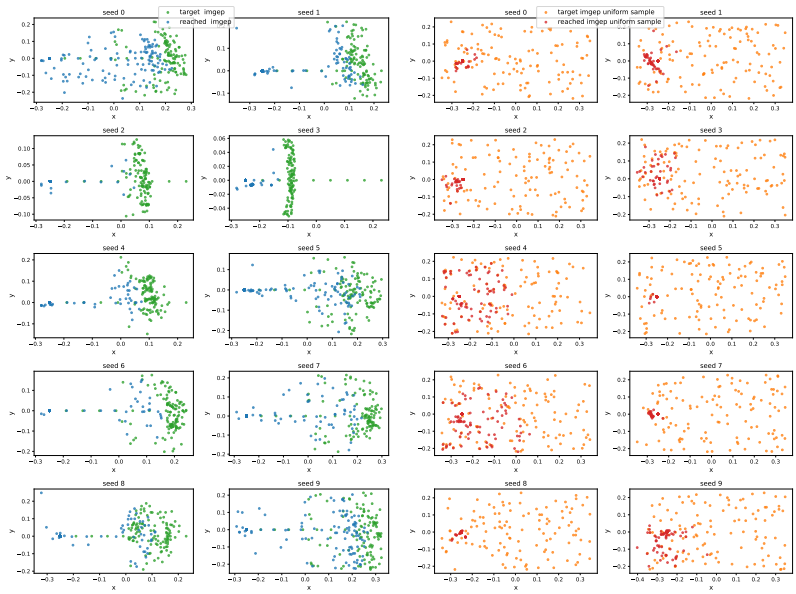


Figure A.10: Target goals and reached positions for every seed of (left) original method (right) IMGEP with uniform sampling of goal. The uniform sampling IMGEP sample a lot of far points that not reached at all

Table A.1: Seed variability

Seed Number	Number of agents	Number of moving (agents)	Number of robust (agents)	max speed (agents)	max speed obs (agents)
Seed 0	107	93	91	2.8	1.4
Seed 1	64	54	26	2.7	1.5
Seed 2	33	32	1	2.0	1.1
Seed 3	18	7	0	0.5	0.3
Seed 4	35	26	6	1.9	0.4
Seed 5	66	52	38	2.9	1.8
Seed 6	54	54	2	2.5	0.3
Seed 7	30	30	1	3.0	0.9
Seed 8	44	44	4	2.3	0.3
Seed 9	104	94	92	3.2	2.3

also emerge with random goal sampling, as the agent will only make progress on goals that either not too far or too close from its current abilities. (see e.g. Forestier et al. 2022 [249]).

A.1.4 Seed variability

We report in tab.A.1 the variability of the results of the method across the 10 seeds. The variability in result might indicate that some parameter area are easier to navigate or more prone to certain behavior. Overall we still observe that every seed finds a good amount of moving agents and most of them find at least 1 robust agent(ie an agent with a score ≥ 0.95 to the "basic obstacle test").

A.1.5 Generalization table

We refer to table A.2 for the full generalization results.

A.1.6 Lenia system

Cellular automata are, in their classic form, a grid of "cells" $A = \{a_x\}$ that evolve through time $A^{t=1} \longrightarrow ... \longrightarrow A^{t=T}$ via local "physics-like" laws. More precisely, the cells sequentially update their state based on the states of their neighbours: $a_x^{t+1} = f(a_x^t, \mathcal{N}(a_x^t))$, where $x \in \mathcal{X}$ is the position of the cell on the grid, a_x is the state of the cell, and $\mathcal{N}(a_x^t)$ is the neighbourhood of the cell. The dynamic of the CA is thus entirely defined by the initialization $A^{t=1}$ (initial state of the cells in the grid) and the update rule f (function that takes a scalar and outputs a scalar,control how a cell updates based on its neighbours). But predicting their long term behavior is a difficult challenge even for simple ones due to their chaotic dynamics.

Lenia is a class of continuous cellular automata (CA) where each CA instance is defined by a set of parameters θ that conditions the CA rule f_θ; once the parameters θ conditioning the update rule has been chosen, the system is a classical CA where the initial grid pattern $A^{t=1}$ will be updated.

In Lenia, the system is composed of several communicating grids $A = \{A_c\}$ which we call channels. In each of these grids, every cell/pixel can take any value between 0 and 1. Cells at 0 are considered dead while others are alive. The channels are updated in parallel according to their own physics rule. Intuitively, we can see channels as the domain of existence of a certain type of cell. Each type of cell has its own physics : it has its own way to interact with other cells of its type (intra-channel influence) and also its own way to interact with cells of other types (cross-channel influence).

The update of a cell $a_{x,c}$ at position x in channel c can be decomposed in three steps. First the cell senses its neighbourhood in some other channels (its neighbourhood in its channel, with cells of the same type but also in other channels with other types of cells) through convolution kernels which are filters K_k of different shapes and sizes. Second, the cell converts this sensing into an update (whether positive or negative growth or neutral) through growth functions G_k associated with the kernels. Finally, the cell modifies its state by summing the scalars obtained after the growth functions and adding it to its current state. After the update of every rule has been applied, the state is clipped between 0 and 1. Each (kernel,growth function) couple is associated to the source channel c_s it senses, and to the target channel c_t it updates. A couple (kernel, growth function) characterizes a rule on how a type of cell c_t reacts to its neighbourhood of cells of type c_s. Note that c_s and c_t could be the same, which correspond to interaction of cells of the same type (intra-channel influence). Note also that we can have several rules, i.e. several (kernel,growth function) couples, characterizing the interaction between c_s and c_t.

A local update in the grid is summarized with the following formula (where G^k, K^k, c_s^k, c_t^k are respectively the growth function, convolution filter, source channel, target channel associated with the k'th rule):

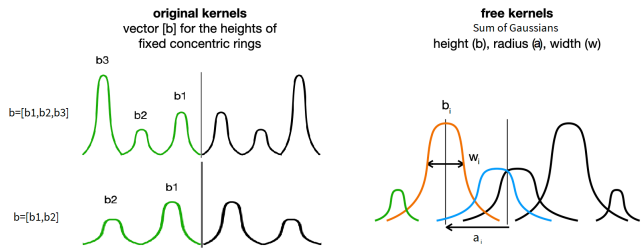


Figure A.11: Visualization of (left) the convolution kernels used in the original lenia papers [8, 9], (right) the kernel we propose in this paper for more differentiation capabilities. The kernel we propose consists of a sum of free shifted gaussian bumps while the one in the original lenia papers consist of fixed concentrated rings.

$$a_x^{t+1} = f(a_x^t, \mathcal{N}(a_x^t)) = \begin{bmatrix} a_{x,c_0}^t + \frac{1}{T}\sum_{k \text{ st } c_t^k=0} G^k(K^k(a_{x,c_s^k}^t, \mathcal{N}_{c_s^k}(a_x^t))) \\ \cdot \\ \cdot \\ a_{x,c_C}^t + \frac{1}{T}\sum_{k \text{ st } c_t^k=C} G^k(K^k(a_{x,c_s^k}^t, \mathcal{N}_{c_s^k}(a_x^t))) \end{bmatrix}$$

For each rule, the shape of the (kernel, growth function) is parameterized. We are thus able to "tune" the physics of the cells and of their interactions by changing the kernels shape (how the cells perceive their neighborhood) as well as the growth function shape (how the cells react to this perception).

Differentiating through Lenia steps

Due to the locality and recurrence of the update rule, there is a close relationship between cellular automata and recurrent convolutional networks [435]. In fact, we can see a rollout in Lenia as applying a recurrent neural network to an initial state. If (some of) the network parameters are differentiable, backpropagation can be done by "unfolding" the Lenia rollout and applying a loss at certain time step(s) like in [257].

However, in the classic version of Lenia, the shape of the kernels are not totally differentiable and not very flexible. To allow easier optimization of the Lenia system, we introduce some changes to the kernel parameterization.

In fact in the original Lenia [8], the number of bumps in the kernel (see Fig.A.11 left) is fixed and cannot be optimized through gradient descent.

We therefore introduced a new class of CA with differentiable parameters. To do so, the main shift is to use kernels in the form of a sum of k overlapping gaussian bumps:

$$x \rightarrow \sum_i^k b_i exp(-\frac{(\frac{x}{rR} - a_i)^2}{2w_i^2})$$

The parameters controlling the shape are then $3k$-dimensional vectors: b for height of the bump, w for the size of the bump and a for the center of the bump.

These symmetric "free kernels", while very inspired from Lenia's original "vanilla bumps", allow differentiation and more flexibility and expressivity but at the cost of more parameters. For example, it is possible to reduce the number of bumps by assigning some null height values, allowing the number of bumps to be optimized through gradient descent.

In Lenia, a growth function $G : [0,1] \to [-1,1]$ is any unimodal non-monotonic function that satisfies $G(\mu) = 1$. In this work, we use the continuous exponential growth function $G(x) = 2\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) - 1$ which is differentiable with respect to μ and σ.

To summarize, the parameters of the update rule are thus those controlling the kernel shape (R, r, a, w, b), those controlling the growth function (μ, σ, h) and a time controlling parameter (T). For a total of n rules (all channels included) with k bumps kernels, the number of parameters is $(3k+4)n+2$. In our experiments, R and T are chosen randomly and fixed while all the other parameters are optimized, and we use a total of $n = 10$ rules with $k = 3$ bumps kernels . So in total we have 132 parameters for the rules from which 130 are optimized.

In addition to the rule, parameters we also optimize the initialization square $I_{square} \in [0,1]^{(40,40)}$.

Obstacles

The multi-channel aspect of Lenia allows the implementation of different types of cells/particles. To implement obstacles in Lenia we added a separate "obstacle" channel with a kernel going from this channel to the learnable "creature" channel (see Fig.2.). This kernel triggers a severe negative growth in the pixels of the learnable channel where there are obstacles but has no impact on other pixels where there are no obstacles (very localized kernel). This way we prevent any growth in the pixels of the learnable channel where there are obstacles. The formula of the growth function is : $G(x) = -clip((x-1e-8), 0, 1) * 10$. Hyperparameters of this handmade rule can be found in A.1.6.

The learnable channel cells can only sense the obstacles through the changes/deformations it implies on it or its neighbours. In fact, as the only kernel that goes from the obstacle channel to the learnable channel is the one we hand-designed, if a macro agent emerges it has to "touch" the obstacle to sense it. To be precise the agent can only sense an obstacle because its interaction with the obstacle will perturb its own configuration and dynamics (i.e. its shape and the interaction between the cells constituting it). This is similar to experiments with swarming bacteria [436], where the swarm agent must learn to collectively avoid antibiotic zones (externally-added obstacles) where the bacteria can't live.

In our implementation, obstacles stay still, meaning that there is no rule that goes toward (and hence no update of) the obstacle channel . As such, an update step in the final system is summarized at the bottom of Fig.2..

To test the agents under moving obstacles, we simply shift the channel of obstacles of a certain amount of pixel at every timestep. This shift of the grid, for an integer value of speed, can be written as a rule of the system from the obstacle channel to the obstacle channel. The rule would be the same on all the grid and is localized as it is a function of the fixed neighbourhood. Moving obstacles with a speed with a rational value (for example 0.5 pixels/timesteps) is done in our case by doing the shift every few timesteps.

Lenia rules parameters

Here is the list of the parameters associated to the rules of a Lenia system with C channels, nb_k rules with kernels with k bumps. We also provide the range used in this work for the learnable channel. In this work we used C=2 channels (one learnable channel and the fixed channel), $nb_k = 10$ learnable rules and 1 fixed rule (for the obstacles).

- Common to all rules
  - T $\in [1, 10]$
- Learnable rules
  - Kernel (convolution filter) parameters:
    - R $\in [15, 40]$ Radius of the kernels (common to all kernels)
    - r $\in [0, 1]^{nb_k}$ relative radius of each kernel.
    - b $\in [0, 1]^{nb_k,k}$ height of the k bumps.
    - w $\in [0.01, 0.5]^{nb_k,k}$ width of the k bumps.
    - a $\in [0, 1]^{nb_k,k}$ position of the bumps on the radius.
  - Growth function $G(x) = 2\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) - 1$ parameters
    - $\mu \in [0.05, 0.5]^{nb_k}$ mean of the gaussian growth function.
    - $\sigma \in [0.001, 0.18]^{nb_k}$ variance of the gaussian growth function.
    - h $\in [0, 1]^{nb_k}$
  - $c_0 = [0] \times nb_k$ source channel (0 is learnable channel)
  - $c_1 = [0] \times nb_k$ destination channel
- Fixed rule
  - Kernel parameters:
    - R = 4 small radius for very localized action
    - r = [1,1,1]
    - b = [1,0,0]
    - w = [0.5,1,1]
    - a = [0,0,0]
  - Growth function $G(x) = -clip((x - 1e - 8), 0, 1) * 10$
  - $c_0 = 1$ source channel (1 is fixed channel)
  - $c_1 = 0$ destination channel

Lenia rule parameter mutations

- Common to all rules

- T : $\mathcal{N}(0, 0.1) \times \mathcal{B}(0.01)$ (mutation then integer)

- Learnable rules
  - Kernel (convolution filter) parameters:
    - R $\mathcal{N}(0, 0.1) \times \mathcal{B}(0.01)$ (mutation then integer)
    - r :$\mathcal{N}(0_{nb_k}, 0.2 \times \mathcal{I}_{nb_k})$
    - b : $\mathcal{N}(0_{3nb_k}, 0.2 \times \mathcal{I}_{3nb_k})$
    - w :$\mathcal{N}(0_{3nb_k}, 0.2 \times \mathcal{I}_{3nb_k})$
    - a : $\mathcal{N}(0_{3nb_k}, 0.2 \times \mathcal{I}_{3nb_k})$
  - Growth function $G(x) = 2\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) - 1$ parameters
    - μ : $\mathcal{N}(0_{nb_k}, 0.2 \times \mathcal{I}_{nb_k}) \times \mathcal{B}(0.1)$
    - σ: $\mathcal{N}(0_{nb_k}, 0.01 \times \mathcal{I}_{nb_k}) \times \mathcal{B}(0.1)$
    - h $\mathcal{N}(0_{nb_k}, 0.2 \times \mathcal{I}_{nb_k}) \times \mathcal{B}(0.1)$

A.1.7 IMGEP details

Algorithm 2: IMGEP pseudo code

Initialization: history $\mathcal{H}$ and models $\mathcal{T}, \Pi, Optim, R$.
for *i=1..N* **do**
  Generate a target goal $\tau_i \sim \mathcal{T}(\mathcal{H})$ `/* use of curriculum learning and diversity search */`
  Train parameters on target goal $\theta_i^* = Optim(\theta_i|\tau_i)$, where $\theta_i \sim \Pi(\mathcal{H}|\tau_i)$ `/* use of gradient descent and stochasticity handling */`
  Evaluate parameters $x_i \sim R(\theta_i^*)$ `/* behavioral characterization */`
  Store in history $H \leftarrow H \cup (\theta_i^*, x_i)$ `/* reuse knowledge for task sampling and training */`
return $\mathcal{H}$

In this section, we first recall the basics of the IMGEP procedure and then go into the details of each element of the method.

Our method described in the pseudo code 2 starts by initializing a pool of (parameters, reached position) couples by random search, this constitutes the initial state of the history $\mathcal{H}$ (details in 7). Then, at each iteration, the method iterates through the following steps (illustrated in Fig. A.12). **1) Sample a new goal** using a goal sampling strategy which takes into account the previously reached positions (details in 7). An example of the sampling distribution can be found in green in Fig 3.a. **2) Infer starting parameters for that goal** by selecting parameters $\{(\theta_l, A_l^{t=1})_i\}_{i=1...t-1}$ associated to a previously reached position in history $\mathcal{H}$ that is close to the sampled goal (details in 11). **3) Optimize parameters toward the sampled goal** by iteratively performing rollouts of the Lenia system under different environmental conditions A_f and applying stochastic gradient descent on the MSE loss between the disk at goal position and the mass of the learnable channel at the last timestep (details in 11). **4) Update history** $\mathcal{H}$ with the newly obtained parameter point and test it in various environmental conditions A_f to estimate its reached position (details in 11)

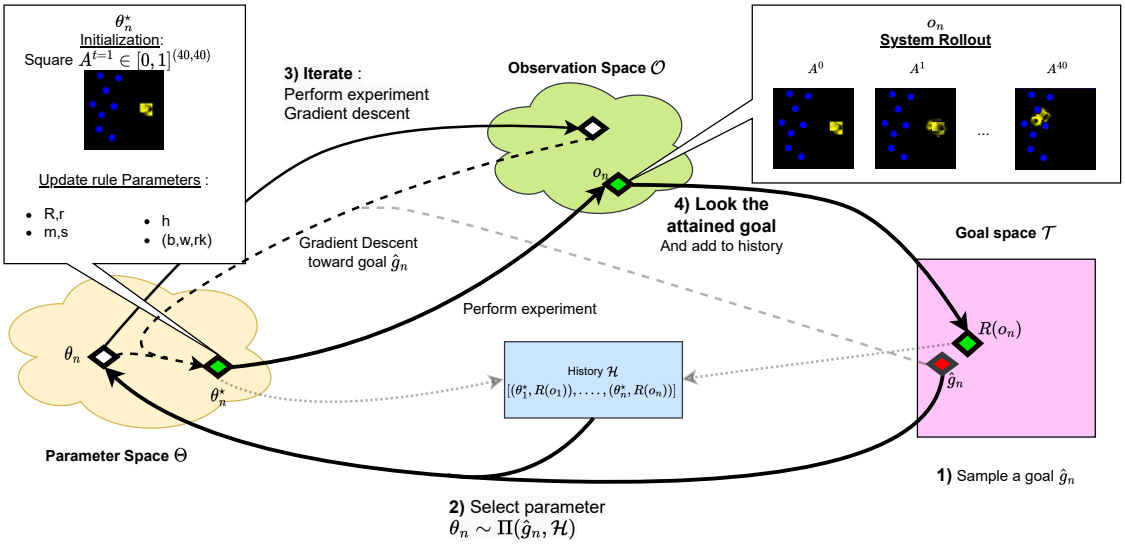


Figure A.12: IMGEP loop

(such that it can be later reused as a starting point for achieving other sampled goals).

As described in the main text, the behavioral space is the position (x,y) of the center of mass at the last timestep of the rollout. The loss we use is the Mean square error loss between the learnable channel at the last timesteps of the rollout and the same grid with a superposition of 2 disk centered at the goal position in the first channel. The target disk has this formula: $0.9x(0.15x(R_g < 10) + 0.85x(R_g < 5))$ where R_g is the euclidian distance to the goal position.

To introduce more diversity in the search (and potentially getting out of difficult optimization landscape), some steps of IMGEP add mutation to the promising parameters before applying the optimization through gradient descent. More details can be found in 11.

Note that we also introduce an automatic way for the method to restart again from scratch in case of not good enough first steps (not present in pseudocode 2). We refer to subsection.7 for a detailed description of this restarting mechanism.

Note that the goal positions as well as the measured reached positions (details in 11) are normalized and centered between -0.5 and 0.5 (so that obstacle positions are at $x > 0$,Fig.3 in the main text) according to the map size SX.

The following sections provide additional details about different parts of the method.

Initialization of history

The IMGEP method first applies an initialization of history $\mathscr{H}$ through random search to bootstrap the whole IMGEP procedure.

In this work, the initialization of history consist of 40 trials of random parameters. The range used for this random search are the one

presented in A.1.6 except that we divide the strength of the kernels parameters h by 3. This change is done in order to have weaker/s-lower updates increasing the chance to have a pattern not exploding or vanishing in 50 timesteps, in order to facilitate further optimization.

This dividing of h by 3 is only to make things go faster (requiring less trials for the initialization of history) with some human heuristic on the system but should not be mandatory as random search without this should also get interesting parameters for initialization with more trials.

Warming up goal sampling

To accelerate the curriculum, we start the first 8 steps of the IMGEP with a deterministic goal sampling which tries to go as far as possible on the x axis. The goal position starts at position (-0.19,0) and is shifted of +0.06 along the x axis for every of those deterministic steps. The rest of the goal sampling is stochastic as described in 7.

Initialization selection

History initialization and the first IMGEP steps have a huge impact on the performance of the method, as it will provide the basis for all subsequent optimization. History initialization and the warm up of goal sampling have a huge impact on the performance of the method, as it will provide the basis for all subsequent optimization.

To mitigate this problem, we also apply initialization selection with the objective of facilitating further optimization. We run the first steps of the method (random initialization and few steps of optimization), and observe the loss for the 3 first deterministic targets (described in section 7). If this loss is above a certain threshold for one of the 3 step, we start over again getting rid of the initialization history and initializing it again with random search. We perform this until we find a "good" initialization that is below the threshold for the 3 steps.

Goal sampling

The goal sampling we chose in this work intends to sample goals ((x,y) positions) that should be most of the time further in the grid (for harder goals), not too far from previously reached positions (for feasibility of the goal) and also not too close from previously achieved goals (to make progress) . From those heuristic we introduce our engineered goal sampling strategy in pseudo code 3. The objective of this engineered sampling is to accelerate the search but much simpler ones could work if given enough computational budget (see ablation with totally random sampling A.1.3).

Algorithm 3: Goal sampling strategy

```
Input: history ℋ nb_close=0,nb_veryclose=0 while nb_close<1
  or nb_veryclose >2: do
    if rand ~ 𝒰(0,1) <0.2 : then
        goal =
            bestgoal(ℋ)+(𝒰(0,1)×0.04+0.02, (𝒰(0,1)×0.45−0.22)/4, )
            /* Try little further than previous
            best                                          */
    else
        if rand ~ 𝒰(0,1) <0.7 : then
            goal=(−𝒰(0,1) × 0.2 + 0.35, −𝒰(0,1) × 0.45 − 0.22) /* T
                */
            ry random far points
        else
            goal=(−𝒰(0,1) × 0.35 + 0.35, −𝒰(0,1) × 0.45 − 0.22)
    nb_close,nb_veryclose=calc_distances(goal,ℋ)
return goal
```

Mutation

We apply mutations on candidates parameters in order to increase diversity. Some mutations can facilitate optimization while others can lead to undesirable configurations impairing it. For this reason, we apply less gradient steps on those mutated parameters. See section 11 for the hyperparameters in this work.

In addition, we generate mutations of a parameter configuration until it results in a pattern not collapsing after 50 timesteps. For this (approximate) collapsing measure, we use a simple soft filter checking if the total mass in the learnable channel at the last timestep is > 10 (to test for death of matter) and if the mean square error between the learnable channel at the last timestep and the disk defined in A.1.7 centered on the center of mass of the learnable channel is < 25 (as a proxy for explosion of the mass, more details in 11). This loop of mutations is counted in the total number of rollout performed by the IMGEP.

We refer to section A.1.6 for the mutation (distribution, mean, variance) applied to each parameters in the method.

Gradient descent

Differentiating through Lenia can be difficult because the gradient must backpropagate through several steps (which moreover have their result clipped between 0 and 1) without vanishing. We should thus limit ourselves to a few iterations when training: in our experiments the loss is applied after 50 steps in Lenia.

Obtaining gradients that are informative for optimization requires an overlapping between the mass in the learnable channel and the disk centered at the goal position. The curriculum we introduce in the goal sampling procedure (7) facilitates this overlap by generating goals

that neither too far nor too close from the initial pattern at t=0 and from previously reached goal.

We refer the reader to appendix section A.A.1.3 for an ablation of the gradient descent showing the importance of it in the method.

Parameter evaluation

We perform an evaluation of the parameters after each IMGEP step (sampling of goal and optimization of parameters). This evaluation consists of running 20 rollouts of 50 timesteps (the same rollout length as in the optimization rollout) with different random obstacle configurations and measures the average reached position over those rollouts.

For each rollout, we also compute the mean square error between the learnable channel at the last timestep and the disk shape centered on the center of mass of the learnable channel at last timestep. We then take the average value over the rollouts. This is used as a proxy "collapsing measure" (explosion or death of the pattern) to apply a soft filter when selecting promising initialization parameter for a new goal as explained in section 11.

The parameters (A_l, θ_l), the measured reached position (r_x, r_y) and collapsing proxy measure c are then stored in the history $\mathscr{H}$.

Reusing history $\mathscr{H}$ for a new goal.

Once a goal is selected, we compute the L2 distance between all vectors (c, r_x, r_y) of the history and (c_{goal}, g_x, g_y), where g_x and g_y are the (x,y) coordinate of the goal and c_{goal} is a constant equal to 0.065 in this work. These L2 distances are used to select a point in the history reaching a position close to the goal while mitigating the risk of collapsing.

In addition to these L2 distances for the selection of potential candidates for a new goal, we also filter out the points in the history having $c > 0.11$ allowing to remove the potential collapsing ones even though they might be close to the goal. We also take into account this collapsing proxy measure as collapsing parameters are hard to recover from through gradient descent.

The candidate parameter for a goal is therefore the point in the history which has $c \leq 0.11$ and which minimize the L2 distance presented above.

IMGEP search Hyperparameters

- Number of IMGEP steps : 120
- History initialization : 40 trials of random parameters.
- In 4 out of 5 IMGEP step, we mutate the candidate parameter before gradient descent.
- Number of gradient steps : 125 when no mutation beforehand (1 out of 5 IMGEP steps) , 15 when mutation beforehand.

- Rollout length : 50 timesteps
- Grid size : 256x256
- Number of obstacle during the search: 8
- Initialization position on the 256x256 grid: [36:76,105:145]

A.1.8 Basic obstacles tests and generalization tests

Note that the tests we provide are proxy measure of agency/stability, and so what we present here are what we consider in this paper as agency. It is for example impossible to test for infinite time stability in finite time budget. Our stability tests are based on previous work on Lenia [234].

Empirical agency test

We describe here the agency test used in the paper:

We first apply a prefilter to the obtained parameters by running a rollout of 500 steps with the obtained parameters. From this rollout, we measure if the mass at the last timestep was strictly above 0 (not dead) and below 6400 (explosion). The number are arbitrary and relatively "loose" so that we reject nearly no "false positive". This prefilter allows to throw out obvious non interesting parameters to reduce the computational cost of testing all obtained parameters – especially for the random search method where many of them are not interesting.

We then do rollout of 2000 timesteps for the empirical agency and moving test. The rollout is long (especially relative to the 50 timesteps of the search) in order to probe for long term stability. We compute some stats, from the rollout observations, which are used for the empirical agency test (and moving test) of the parameters inspired by [234].

The empirical agency test consist of :

- Measuring if the mass of the learnable channel is > 0 and < 6400 (~10% of the map) at the last timestep of the rollout as those correspond to collapse and explosion.
- Measuring if the average mass is augmenting or decreasing too much between 2 windows of the rollout. This is a proxy measure for long term instability meaning that a big loss or increase of mass between the 2 windows is most of the time an indicator for long term instability. In this work, we measure the ratio between the average mass during the 0 to 500 window and 1500 to 2000 window. If this ratio is greater than 2, the parameters do not pass the test. The windows are relatively large to still allow for variation of mass during a rollout and the formation of a pattern in the first window.
- We also want the emerging pattern to be a spatially localized **Soliton** (ie pattern forming a single entity not expanding indefinitely, with a bounded radius). To measure this, we perform a

connectivity analysis of the pattern depending on the kernel radius, rejecting patterns where two distinct blobs of mass cannot influence each other (distance between blobs $\geq R * max(r)$).

Moving test

To test if a pattern passing the empirical agency test is moving, we measure if the center of masse of the learnable channel moved further than 100 pixels from the initialization position at any point during the 1000 first steps of the rollout.

Speed measure

To measure speed of agents, we use the 2000 timesteps rollout computed in the filter phase and track the average distance travelled by the center of mass of the agent on sliding overlapping windows of size 25 starting from timestep 150 to timestep 2000. The result is divided by 25 (the size of the sliding window) in order to have a per timestep average distance travelled. We use a sliding window to filter slight back and forth movement of the center of mass (which can even be due to self organization without clear "movement" of the whole). Note that we compute the speed only for agents passing the filters above.

The same is done to measure speed with obstacles but we average on the 50 rollouts with random obstacles computed in the robustness test. The only small modification is that if an agent does not pass the survival tests above on the rollout (for example its mass reaches 0), we set the speed for this rollout to 0.

Basic obstacles tests

We then test the parameters leading to moving agents by performing 50 rollouts of 2000 timesteps where obstacles are the same as in training i.e. obstacles of radius 10. We place 24 obstacles in the whole grid (compared to only the right part of the grid in training), from which 23 are randomly placed and one being in the trajectory of the moving agent to be sure that it will encounter at least one obstacle in the rollout. To do this we look at the achieved position of the moving agent without obstacle at timestep 1000 and put an obstacle here in the test for every rollout. We also remove any obstacle pixel in the initialization area (pixel of the learnable channel > 0 at the initialization) as well as in a radius of 10 pixels (euclidian distance) of the initialization (to let some space for the initialization to develop).

From the observations of the rollout we compute the same statistics and same categories used for the agency test. To get the robustness measure we then measure the fraction of rollout where the pattern pass the empirical agency test.

Generalization tests

Here is a full description of each of the generalization test conducted in the *Generalization* section in the main text. For all the quantitative generalization tests, we used the same robustness test as above except that we do it on 10 random trials instead of 50: we run rollout of 2000 timesteps, then measure if it fulfills the empirical agency test. The measure of robustness is again measured by the proportion of trials where the agent pass the empirical agency test. (hence between 0 and 1).

We also provide a more detailed table of generalization results in Tab.A.2 adding also agents obtained through random search and semi-manual search.

- **Initialization noise**. In this experiment, we add a centered gaussian noise to the pixel of the initialization square A^1. In the first test "init noise rate" we vary the proportion of pixels affected by this gaussian noise, testing proportions [0.2,0.4,0.6,0.8,1.], and keep the variance fixed to 1. In the "init noise std" test, we apply the noise to all pixels of the initialization but vary the variance of the gaussian in [0.5,1.5,2.5,3.5,4.5].
- **Obstacles** In all of these test we also remove obstacles pixel from the initialization square and in a radius of 10 pixels (euclidian distance) around it.
  - **Obstacle radius** In this test, we vary the radius of the obstacles in [4,7,10,13,16]. The number of obstacles varies according to the radius of obstacles to keep the same ratio of obstacle pixels with the default one which is 24 obstacles of radius 10. The formula is Number obstacles $= 24\times(10/var)^2$.
  - **Obstacles number** In this test, we vary the obstacle number keeping the radius fixed to the default one (radius=10). We try obstacle number= [24, 30,36,42 ,48] .
  - **Obstacle speed**. In this test, we change the dynamic of the obstacle channel so that obstacle move at a certain speed as detailed in A.1.6. For a speed of 1, the obstacle channel is shifted of 1 on the left at every timestep, for a speed of 0.5, the obstacle channel is shifted of 1 every 2 timesteps. We tested obstacle speed of [1/3,1/2,1,2,3]. In this test we put 24 obstacles of radius 10.
- **Scale** In this test, we vary the scale of agents by changing their kernel size multiplying the parameter R of the simulation by the factor. A smaller (resp bigger) size of kernel means that the convolution will cover a smaller (resp bigger) neighbourhood. We also change the initialization size by a factor α to match the scale. To do this, we use a downscaling (or upscaling) of the initialization 40×40 square with bilinear interpolation. We test both smaller sizes : 0.15,0.65 , as well as bigger sizes: 1.15,1.65,2.15.
- **Update**. In this tests, we perturb the update (what is added to the current state) from step 0 until step 1900. We let the step from 1900 to 2000 free of update perturbation to allow the rule to recover until step 2000 for the statistics computation.
  - **Update mask** In this test, for a value of update mask $p < 1$, every pixel has a probability p of being updated while the

rest of the pixels will keep the same value. This does not apply to the update applied by the obstacles. For a value $1 < p < 2$, each pixel is updated one time using the update rule normally (sensing and add of growth) giving a new state and then each pixel is updated again from this new state with a $p - 1$ probability (the sensing on the potential second random update is done by sensing the new state). We test the update mask rate in [0.2,0.6,1.,1.4,1.8].

- **update noise std** In this test, we add noise to the update of the learnable channel before the clipping as such :

$$A_l^{t+1} = A_l^t + \frac{1}{T}\left(G(K * A^t) + \mathcal{N}(0_{256\times256}, \sigma \mathbb{I}_{256\times256})\right)$$

  where $\mathcal{N}(\mu, \Sigma)$ is a gaussian vector of mean μ and variance Σ. We vary σ in $[0.5, 1.5, 2.5, 3.5, 4.5]$
- **Update noise rate**. We add noise to the update of the learnable channel before clipping. Every pixel has a probability $p \in [0.2, 0.4, 0.6, 0.8, 1.]$ to have a gaussian noise of mean 0 and variance 1.

- **Morphological computation/ Hand damage**. In this test, we allow an exterior experimenter to pause the simulation and put pixels of the learnable channel to 0. After the damage, we then let the simulation unroll as usual starting from the damaged state $A_l^{damaged}$.
- **Interactions (Multi agents setting)**. We allow to put several initialization square in the learnable channel. As the update rule apply to all the grid the same way, if a couple (initialization square, update rule) already led to a an agent in the case of a single initialization square then several of them that are not interfering (further enough so that the convolution of a pixel of one does not contains pixels of the other) will lead to several agents.
- **Custom obstacles**. We allow an experimenter to freely draw obstacle in the grid. This allows to have obstacles with shapes not seen during training.
- **Custom init states** In this test, we replace the initialization of the pattern (that was optimized) by simple arbitrary shape such as disk with a gradient (the gradient being to have an asymmetry for movement), disk of large size etc. The web demo at http://developmentalsystems.org/sensorimotor-lenia-companion also allows to load any image as initialization of the system.
- **External control** This experiment consists in adding a new channel (a new type of cell) to the system which we want to act as an attractive element. We conducted a semi handmade search in order to search for a rule, sensing in the attractive channel and updating the learnable channel, leading to this attractive behavior.
  Note that this attractive element should attract but not disturb too much the matter as we don't want the attractive matter to be able to destroy the agent dynamics.
  In fact, we first searched for a rule tuned for one agent found with the IMGEP search (ie one parameter point (A_l, θ_l)). By doing so, the rule is adapted to the dynamic of this specific agent (for

example different agents might have different range for pixel value or growth etc).
The search for a rule (tuned for a specific agent) is semi handmade. We first preselect some rule parameters from a set of random rules. The preselection is done by moving a circle of attractive mass along a predefined straight trajectory in an environment with a moving agent. We then look if the attractive mass and the agent overlaps at the last timestep which should mean that the agent followed this attractive mass. An experimenter then select by hand the rules that lead to attraction of mass without too much perturbation by controlling the mass of attractive matter in a real time simulation with the moving agent.
After searching for a rule for a specific agent, we then tested it on some other moving agents obtained with IMGEP. Some agents (some parameters (A_f, θ_f)) are more prone to work with it (meaning attraction while not affecting the stability too much) while it destroy the stability of others. The reported qualitative results on this test are performed on agents where the rule leads to stable attraction.

A.1.9 Comparison baselines

Random search details

We use uniform sampling of parameters with the ranges given in A.1.6.

The initialization 40x40 square is randomly sampled with each of the pixel constituting it being independently sampled following a uniform distribution between 0 and 1.

"Handmade" agents (from original Lenia paper)

The parameters from this dataset are the one from the original Lenia paper [8, 9] (following these links: https://github.com/Chakazul/Lenia/tree/master/Python/found, and https://github.com/Chakazul/Lenia/blob/master/Python/old/animals.json).
Contrary to the rest of the paper we use the classic parameterization of Lenia for the agent channel. We filter out those that have more than one channel or an initialization that has a side bigger than 256. We then apply the pre-filter and filter as explained in section A.1.8. We provide the resulting parameters in the data folder of https://github.com/flowersteam/sensorimotor-lenia-search.

In the handmade search from the original Lenia papers, self-organizing patterns were discovered by basic evolutionary algorithms, through one of these routes: (1) random parameter values and initial patterns; (2) start from an existing moving pattern and mutate the parameter values; (3) manual editing of the initial pattern.

A.1.10 Movie legends

You can find all movies on this companion website https://developmentalsystems.org/sensorimotor-lenia-companion/.

- **Movie S1: Sensorimotor agents** Different agents (yellow) emerging from rules obtained by the IMGEP. The agents display sensorimotor capabilities: they are robust and react to perturbations by the obstacles (blue). The righmost video shows the system with a different colormap (fixed obstacle channel in black) to highlight the differences in activity in the agent as a response to perturbation.
- **Movie S2: Random search** Each 100 squares are random parameters trials (each 1 channel and 10 rules so 130 parameters for all the rules of a square). We observe that a lot of random search trials lead to death or explosion of the mass. Very little lead to stable spatially localized pattern and even less to moving ones.
- **Movie S3 Orbium, moving agent from the original lenia papers, fragile to external perturbations** S3.a: Orbium: the equivalent of the glider in Lenia (from the original lenia paper), an example of moving agent. S3.b and S3.c videos: collision between several orbium leading to death/explosion. This shows the fragility of the orbium to external perturbations.
- **Movie S4 Orbium perturbed by obstacles** Orbium, equivalent of the glider in Lenia (from the original lenia paper), dies from perturbations by obstacles.
- **Movie S5: Agents obtained by each method** 100 Patterns passing our agency tests obtained by each method: random search(S5.a),IMGEP (S5.b), handmade search ((S5.c)from Lenia original papers). A lot of IMGEP obtained agents are moving agents with high speed while a lot of agents obtained by random search are static.
- **Movie S6: Moving obstacle test on agents obtained by each method** 100 Agents obtained by random search(S6.a),IMGEP(S6.b), handmade search ((S6.c) from Lenia original papers). We observe that the proportion of agents with robustness to moving obstacles is much higher in the agents obtained by IMGEP than the ones obtained by random search and handmade search.
- **Movie S7: Illustration of the quantitative generalization tests performed** See companion website https://developmentalsystems.org/sensorimotor-lenia-companion/. Videos of quantitative tests for 2 moving agents obtained by IMGEP. We display only a subset of the value tested for every quantitative test.
- **Movie S8: Out of distribution obstacles: Different shapes** Test of a moving agent obtained by IMGEP on obstacles that were not seen during training.
- **Movie S9: Out of distribution obstacles: maze.** Test of a moving agent to maze like obstacles.
- **Movie S10: Out of distribution obstacles: Bullet like obstacles** Test of a moving agent to bullet like environment: fast small moving obstacles.
- **Movie S11: Individuality preservation** Example of moving agents obtained by IMGEP colliding while keeping their individuality, they don't merge or collapse from the collision.

- **Movie S12: Reproduction** For some moving agents, under specific conditions, the collision of 2 agents can lead to the self-organization of a 3rd agent. (each with its own individuality)
- **Movie S13: Attraction** Example of moving agents attracting each other while still maintaining their own individuality.
- **Movie S14: Asynchronous update** Testing a moving agent with asynchronous updates. Each cell is updated with a certain probability at each step leading to cells being asynchronously updated.
- **Movie S15: Scaling the agents down** The moving agents size is reduced. The scaled down agents still seem to behave similarly (same shape and have sensorimotor capabilities) to the normal size one while being composed of less cells.
- **Movie S16: Morphological computation** We pause the simulation and remove some cells of a moving agent. As a response to this alteration of the structure, the moving agent changes direction, regrow itself and moves away. This video isolates the fact that the macro agent senses perturbations of its structure and respond to it by a morphological growth.
- **Movie S17: External control.** We introduce an attractive element in another channel (in Cyan). We learned the rule that control the way this external element channel influences the learnable channel(Yellow) and display the resulting behavior here. The moving agent is effectively attracted to this introduce component. By controlling the external element we can control live the direction of the moving agent.
- **Movie S18: Robustness to initialization** Testing the robustness of the learned rule to emerge an agent from different initial patterns. We replace the learned initial pattern by : S18.a a disk with a gradient; S18.b a large disk (much larger than an agent); S18.c top a disk with gradient of another size, bottom a disk without gradient. Some initialization lead to the robust emergence of one or several agents while some lead to the collapse of the pattern.
- **Movie S19: Examples of agents considered non moving by our moving test**

Table A.2: Generalization results

Tests	IMGEP		Random	Handmade
	speed> 1	10 best	10 best	10 best
speed	1.33 ± 0.28	1.94 ± 0.15	0.53 ± 0.25	0.34 ± 0.10
obstacle				
number				
24	0.98 ± 0.07	0.99 ± 0.03	0.99 ± 0.03	0.99 ± 0.03
30	0.98 ± 0.07	1.00 ± 0.00	0.99 ± 0.03	0.99 ± 0.03
36	0.99 ± 0.06	1.00 ± 0.00	0.99 ± 0.03	0.97 ± 0.09
42	0.99 ± 0.03	1.00 ± 0.00	0.99 ± 0.03	0.97 ± 0.09
48	0.99 ± 0.04	1.00 ± 0.00	1.00 ± 0.00	0.98 ± 0.06
radius				
4	0.92 ± 0.18	0.90 ± 0.13	0.92 ± 0.12	0.95 ± 0.09
7	0.98 ± 0.08	1.00 ± 0.00	1.00 ± 0.00	0.97 ± 0.09
10	0.98 ± 0.07	0.99 ± 0.03	0.99 ± 0.03	0.99 ± 0.03
13	0.98 ± 0.08	0.99 ± 0.03	1.00 ± 0.00	0.99 ± 0.03
16	0.98 ± 0.08	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
speed				
1/3	0.99 ± 0.04	1.00 ± 0.00	0.77 ± 0.27	0.74 ± 0.28
1/2	0.97 ± 0.07	1.00 ± 0.00	0.61 ± 0.38	0.51 ± 0.38
1	0.81 ± 0.23	0.97 ± 0.05	0.42 ± 0.41	0.02 ± 0.04
2	0.34 ± 0.32	0.71 ± 0.25	0.13 ± 0.29	0.00 ± 0.00
3	0.12 ± 0.15	0.32 ± 0.17	0.07 ± 0.12	0.00 ± 0.00
update				
mask rate				
0.2	0.99 ± 0.08	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
0.6	0.99 ± 0.08	1.00 ± 0.00	0.89 ± 0.30	1.00 ± 0.00
1.0	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
1.4	0.99 ± 0.09	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
1.8	0.99 ± 0.10	1.00 ± 0.00	1.00 ± 0.00	1.00 ± 0.00
noise rate				
0.2	0.91 ± 0.28	0.90 ± 0.30	0.77 ± 0.37	0.99 ± 0.03
0.4	0.75 ± 0.42	0.91 ± 0.27	0.74 ± 0.38	0.92 ± 0.18
0.6	0.67 ± 0.45	0.90 ± 0.27	0.58 ± 0.46	0.77 ± 0.38
0.8	0.60 ± 0.47	0.63 ± 0.44	0.50 ± 0.44	0.71 ± 0.44
1.0	0.51 ± 0.47	0.32 ± 0.41	0.44 ± 0.45	0.70 ± 0.46
noise std				
0.2	0.99 ± 0.11	1.00 ± 0.00	0.96 ± 0.12	1.00 ± 0.00
0.6	0.79 ± 0.39	0.90 ± 0.30	0.76 ± 0.39	0.98 ± 0.06
1.0	0.51 ± 0.47	0.32 ± 0.41	0.44 ± 0.45	0.70 ± 0.46
1.4	0.08 ± 0.21	0.03 ± 0.09	0.18 ± 0.32	0.56 ± 0.45
1.8	0.06 ± 0.14	0.06 ± 0.10	0.17 ± 0.30	0.45 ± 0.47
init				
noise rate				
0.2	1.00 ± 0.01	1.00 ± 0.00	0.89 ± 0.16	1.00 ± 0.00
0.4	0.99 ± 0.09	1.00 ± 0.00	0.91 ± 0.24	0.99 ± 0.03
0.6	0.98 ± 0.13	1.00 ± 0.00	0.88 ± 0.30	0.95 ± 0.15
0.8	0.97 ± 0.14	1.00 ± 0.00	0.88 ± 0.30	0.89 ± 0.24
1.0	0.95 ± 0.21	1.00 ± 0.00	0.88 ± 0.30	0.76 ± 0.29
noise std				
0.5	0.97 ± 0.16	1.00 ± 0.00	0.87 ± 0.30	0.97 ± 0.09
1.5	0.94 ± 0.20	0.98 ± 0.06	0.85 ± 0.30	0.52 ± 0.42
2.5	0.89 ± 0.27	0.92 ± 0.17	0.80 ± 0.36	0.37 ± 0.44
3.5	0.86 ± 0.32	0.91 ± 0.27	0.81 ± 0.34	0.35 ± 0.45
4.5	0.85 ± 0.32	0.94 ± 0.18	0.79 ± 0.38	0.32 ± 0.43
scaling				
0.15	0.91 ± 0.28	0.90 ± 0.30	0.30 ± 0.46	0.00 ± 0.00
0.65	0.99 ± 0.10	1.00 ± 0.00	0.50 ± 0.50	1.00 ± 0.00
1.15	1.00 ± 0.00	1.00 ± 0.00	0.70 ± 0.46	1.00 ± 0.00
1.65	1.00 ± 0.00	1.00 ± 0.00	0.70 ± 0.46	1.00 ± 0.00
2.15	1.00 ± 0.00	1.00 ± 0.00	0.60 ± 0.49	1.00 ± 0.00

A.2 Appendix: Flow-Lenia: Towards open-ended evolution in cellular automata through mass conservation and parameter localization

A.2.1 Details on the optimization procedure

In this section, we provide details about the optimization of flow lenia creatures (Sec.1.3.5).

We used evosax lange2022 implementation of the OpenES strategy with population size of 16 and adam optimizer kingma2017 with 0.01 as learning rate. We optimized the Flow Lenia update rule with different number of kernels and either 1 or 2 channels. For comparison, we also trained original Lenia on the directed motion task following the same optimization procedure. The initial pattern is composed, as in random search, of a square patch with non-zero activations placed at the center of the world and zeros everywhere else. Results are shown in figure 1.19.a and code used to run experiments is available at this link.

Directed motion

In order to train creatures displaying directed motion, i.e straight line motion, we used the distance travelled by the creature as the fitness function. The distance is calculated by computing the center of mass of the pattern at step 0 and final step 400. Formally, the fitness function is defined as :

$$f(\theta) = dist(\phi(A^0), \phi(A^{400}))$$

Where $A \equiv \{A^0, ..., A^T\}$ is the pattern obtained by making a rollout with parameters θ for T timesteps (here 500). $\phi(A^t) \in [-0.5, 0.5]^2$ is the center of mass of state A^t and $dist$ is the euclidean distance function. We optimized the system with either 1 or 2 channels and 10 or 20 kernels. We used $M = \begin{bmatrix} 5 & 5 \\ 5 & 5 \end{bmatrix}$ as the adjacency matrix with 2 channels and 20 kernels and $M = \begin{bmatrix} 3 & 2 \\ 2 & 3 \end{bmatrix}$ with 10 kernels.
Results (see fig 1.19.a) show that good solutions can be found in the 2 channels condition but not in the single channel case. However, when running the algorithm for longer (e.g 5000 generations), we have been able to found single channel creatures with similar fitness than their 2 channels counterpart. Increasing the number of kernels led to faster discovery of good solutions. The best performing creature is shown in figure 1.19.b . This creature moves because of attraction/repulsion dynamics between the 2 channels which might explain why directed motion is much easier to attain with multi-channels creatures. On the other hand, the optimization of the original Lenia model is much less

stable, and discovered patterns are less successful than their mass-conservative counterparts. Moreover, every Lenia optimized patterns are exploding ones.

Angular motion

In this task, we want emerging creatures to display more complex forms of motion. More precisely, we want creatures to be able to move and make turns. As with directed motion, we use the center of mass of the creature through time to compute its trajectory. The fitness function is the following :

$$\begin{aligned} f(\theta) = & \, dist(\phi(A^0), \phi(A^{200})) \\ & + dist(\phi(A^{200}), \phi(A^{400})) \\ & + \angle[\phi(A^{200}) - \phi(A^0)]\,[\phi(A^{400}) - \phi(A^{200})] \end{aligned}$$

Where $\angle ab$ is the angle between vectors a and b. The first two terms are the distance travelled from step 0 to step 200 and from step 200 to step 400. The last term is the angle between these two trajectories which is maximal when they are opposite. In order to avoid large angles to come from very small movements, the angle is set to 0 when distance traveled either before or after step 200 is below a given threshold. The optimal behavior for this fitness function is then to move fast in one direction, make a 180° turn, and then move fast in the opposite direction. We used 2 channels, 20 kernels and the same connectivity matrix as for directed motion.
Result are shown in figure 1.19.a. The best performing creature, shown in figure 1.19.c, displays very complex internal dynamics leading it to periodically make 180° turns while moving in straight line the rest of the time. These dynamics seem to be generated by attraction repulsion dynamics like the ones observed in directed motion but here in a more intricate morphology.

Navigation through obstacles

In this task, we want to see if creatures can navigate through obstacles as done in contribution Sec.1.2. To do so, we added walls which are implemented by adding a strong flow going from the center of walls outwards, thus strongly repelling the creature and acting as a solid obstacle. At each evaluation of the optimization process, we randomly sample points on a circle surrounding the creatures' initial positions to be walls positions thus making a "forest" of walls around the creature. We then optimize the creature with the same fitness function as in the *directed motion* task so creatures have to go as far as possible and so through the forest. We made the experiment with 2 channels creatures, walls are defined in a separate third channel. We used 25 kernels and $M = \begin{bmatrix} 5 & 5 & 0 \\ 5 & 5 & 0 \\ 5 & 0 & 0 \end{bmatrix}$ as the connectivity matrix so creatures are able to sense the walls channel (3rd channel).

We have been able to successfully train creatures able to move and stay robust when making contact with walls such as the one shown in figure 1.19.d which is able to resist deformation and find a way out the "forest". In comparison, solving a similar task in Lenia required complex optimization methods based on curriculum learning, diversity search and gradient descent over a differentiable CA in Sec.1.2. However, such a comparison is difficult because Flow Lenia creatures are inherently more robust due to conservation of mass, whereas Lenia creatures can disappear because of perturbations.

Chemotaxis

Another important feature of natural life-forms is the ability to sense their environment in order to find food or avoid dangers through chemotaxis. In this task, we want creatures to be able to sense a "chemical" gradient and climb it towards its maximum. To do so, we added a separate channel $\Gamma : \mathscr{L} \to \mathbb{R}_{\geq 0}$ whose activations are defined following a Gaussian function around a point randomly sampled on a circle surrounding the center of the CA for each evaluation of the optimization process ensuring creatures learn to follow the gradient and not a fixed direction while keeping the distance to cover constant. We also added 5 kernels and growth functions from Γ to A, which are also optimized, so the creature is able to sense the chemical. The fitness of an individual is then computed with the following function :

$$f(\theta) = \frac{\sum_{x \in \mathscr{L}} A_{\Sigma}^{500}(x) \times \Gamma(x)}{\sum_{x \in \mathscr{L}} A_{\Sigma}^{500}(x)}$$

Since mass is conserved, the optimal behavior for a creature is to concentrate as much of its mass in the cells where Γ is maximal.
We have been able to find good solutions to this task as shown in figure 1.19.a . Best solutions such as the one shown in figure 1.19.e are perfectly able to climb the gradient towards its maximum.

A.3 Appendix: Eco-evolutionary Dynamics of Non-episodic Neuroevolution in Large Multi-agent Environments

A.3.1 Details of the simulation

Environment

Our simulation environment is an extension of the CPR environment [300, 301] that the AI community has been using to study the emergence of cooperation in groups of self-interested agents: a two-dimensional grid-world where some cells contain resources (in green) that the agents (in black) can collect. Resources grow depending of the presence of other resources around them, which means that there is a

positive feedback loop, with reduction in resources leading to further reductions. In addition to resources, the environment may contain walls (in blue) that kill agents trying to traverse them (see Figure 2.3 for an illustration of our environment).

At each time step t of the simulation a resource may grow in a pixel in a cell of the environment with location (x, y) based on the following three processes:

- a neighborhood-dependent probability $p_I(x, y)$ determines the probability of regrowth in a cell based on the number of resources in its neighborhood, I
- a niche-dependent scaling factor $c(x)$ is used to scale p_I. We employ a latitudinal niching model used in previous studies [302, 312]: the world is divided into N niches, each one having the form of a horizontal stripe of pixels so that a cell's location depends only on its vertical position x. We refer to $c(x)$ as the climate value of niche x.
- independently of its neighbors and niche, a resource grows with a constant low probability c. This is what we refer to as (sparse) spontaneous growth.

By modeling resource generation in this way we ensure that the resource distribution follows the CPR model, that it exhibits additional spatio-temporal variability due to the presence of niches and that resources do not disappear too easily, which can be problematic in reset-free environments. Thus, the combined regrowth rate for a resource r is:

$$p(x, y) = p_I(x, y) \cdot c(x) + c \tag{A.1}$$

A niche's climate value is determined by equation: $c(x) = (\alpha^x + 1)/(\alpha + 1)$, which returns values from 0 to 1 and allows us to control the relationship between niche location and climate to be from linear to exponential.

The agents

At each time step there is a variable number of agents K_t in the environment, each one characterized by its sensorimotor ability, cognitive capacity and physiology.

Sensorimotor ability An agent observes pixel values at each time step within its visual range (a square of size $[w_o, w_o]$ centered around the agent, as illustrated in the bottom right part of Figure 2.3). The pixel values contain information about the resources, other agents (including their number) and walls. At each time step an agent can choose to stay inactive or execute an action to navigate up, down, right or left.

Cognitive capacity An agent is equipped with an artificial neural network that outputs the action to undertake based on the current observation and whose weights are initialized randomly once at the start of the simulation. Its architecture (illustrated in the bottom right part of Figure 2.3) is minimal: a convolutional neural network, an LSTM cell that equips the agents with memory by enabling policies conditioned on a trajectory of observatories and a linear layer that transforms hidden states to actions.

Physiology An agent is equipped with a simple physiological model modulating its level of energy: the agent is born with an initial energy value E_0 which, at every time step, experiences a linear decrease, and, if the agent consumes a resource, is increased by one (see the top right part of Figure 2.3 for an illustrative example of how the energy level may change within the lifetime of a hypothetical agent). The energy is also clipped to a max value E_{max}.

Non-episodic neuroevolution

In neuroevolution (NE) a population of neural networks adapts its weights through random mutations and a selection mechanism that promotes well-performing policies. Under a classical NE paradigm training time is divided into generations, at the end of which agents reproduce to form the next generation [99, 304].

Our proposed system deviates from this paradigm in two respects:

- agents do not reproduce according to their fitness but according to a minimal criterion [303, 437] on their energy level;
- evolution is non-episodic: upon satisfying certain criteria an agent reproduces locally (the off-spring appears on the same cell as its parent), so that agents are added in an online fashion to the population, removing the need for a concept of generation.

Reproduction In order to reproduce an agent needs to maintain its energy level above a threshold E_{min} for at least T_{repr} time steps. Once this happens the agent produces an off-spring and is a candidate for reproduction again. Thus, agents may have a variable number of off-spring and do not die upon reproduction. We illustrate this relationship between energy level and reproduction in the top right part of Figure 2.3. Reproduction is asexual: an agent's weights are mutated by adding noise sampled from $\mathcal{N}(0, \sigma)$

Death An agent dies once its energy level has been below a threshold E_{min} for at least T_{death} time-steps or if its age is bigger than a certain value L_{max}. Once this happens, the agent is removed from the population forever.

Hyperparameters of the simulation

Below we provide the values of all hyper-parameters of the environment:

- grid size : 400x200
- Max population :1000 agents
- Starting population : 330 agents randomly placed
- Starting resources : 16 0000 randomly placed resources (there can only be at most 1 resources at one pixel)
- Field of view of agents w_o : 15 (15x15 square with 7 in every direction)
- Total number of timesteps : 1e6
- Mutation variance σ : 0.02
- Energy function parameters :
  - Time to reproduce $T_{reproduce}$: 140 timesteps
  - Time to die T_{death}: 200 timesteps
  - Max energy E_{max} = Starting energy E_0 : 3
  - Energy death E_{death} : 0
  - Energy decay : 0.025
  - Increase in energy when eating a resource : 1
  - Maximum age : 650
- Regrowth function
  - $p(x, y) = p_I(x, y) \cdot c(x) + c$
  - $p_I(x, y) = \mathbb{1}_{I=1} * 0.002$
  - I corresponds to the 4 direct neighbors
  - $c(x) = \frac{(\alpha^x+1)}{(\alpha+1)}$ with α = 200
  - c = 0.00005

The simulation of the environment with a large number of agents and 1e6 timesteps took 20 minutes on a single GPU thanks to JAX parallelization and speedup.

Details of agents architecture

The observation of the agent is fed into a 2 layer Convolutional neural network (CNN) : First CNN has a kernel size (3,3), stride 2 and number of features 4 followed by an average pooling of size (2,2) and stride 1 , the second CNN has a kernel size (3,3, stride 2 and number of features 8 followed by an average pooling of size (2,2) and stride 1. The output of the CNN is then flattened and we concatenate to this vector the previous action of the agent as well as a binary telling if the agent has eaten a resources or not. This embedding is then fed into the LSTM of hidden state size 4 and we concatenate the embedding with the output of the LSTM. The vector obtained is then fed into a dense layer of size 8 and a tanh activation function. We finally apply a last denser layer of size 5 (the number of actions) with softmax to get the action probability from which the action will be sampled. The softmax we use has a low temperature (1/50), so that the evolution can quickly learn non random policy.

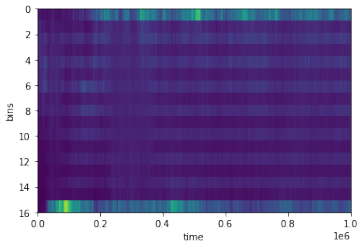


Figure A.13: Heatmap of the amount of individuals in 17 bins of the distance traveled during a time window in seed 1. y axis corresponds to the bins while x axis is the timesteps of the natural env simulation. On the bin (y axis) 0 are agent that traveled a very small distance while 16 are agent that covered a long distance (nearly the maximum amount of distance you can travel during the 50 timesteps window)

In total the agent neural network is very small with only 2445 parameters. We chose a small neural network in order to make the evolution learning easier.

A.3.2 Details on measures and evaluation

The metrics used to characterize the system in the natural and lab environment are:

- Amount of resources in the map : sum over the whole grid of the resource channel
- Population size: Number of individual alive in the simulation
- Expectancy : Average of age of agent that died during a large time window of 500 timesteps.
- Percentage of the population with different amount of movement : For every agent alive during a time window of 50, we compute the Manhattan distance between the position at the last timestep of the window and the position at the beginning of the window. This gives the distance traveled during this time window for every agent alive as a number between 0 and 50. We then make 17 bins out of those 51 possible distance traveled. We then report in Figure 2.4.B the percentage of the population in the two extreme bins as well as the max on the other ones. We provide in Figure A.13 a heatmap displaying the all 17 bins during the whole natural environment simulation of seed 1.

For statistical testing we employ the ANOVA test to detect differences across multiple conditions and Tukey's range test for pairwise comparisons. We report as statistically significant differences between pairs of methods with a p-value lower than 0.05.

A.3.3 Additional results

Large-scale trends

At the large scale several phases can be seen in the evolution of agents and the environment. In this section, we explore those phases in seed 1 (which is the seed described in the main paper in section 2.1.4) and give some new metrics of the natural environment not mentioned in the main paper such as mean life expectancy of the agents.

Population size and life expectancy rise and plateau

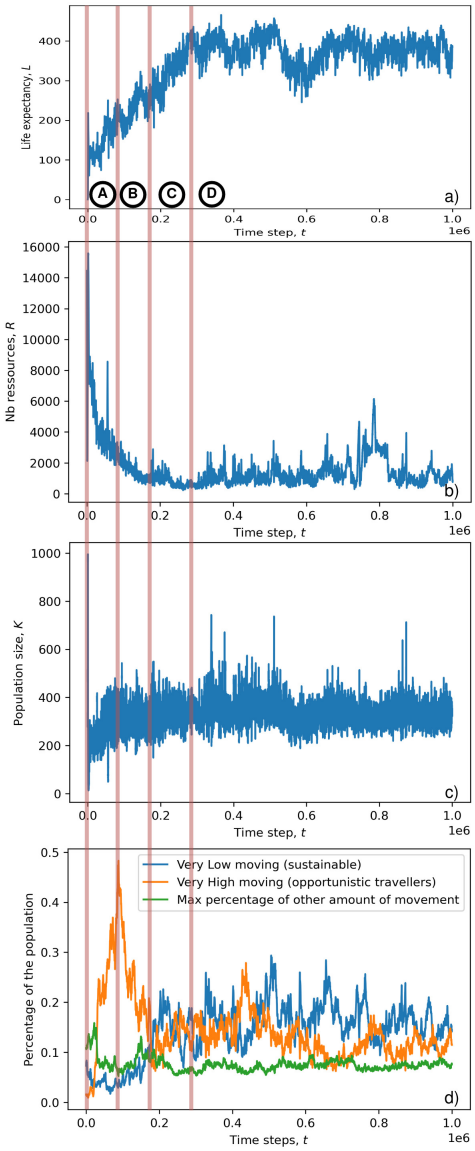


Figure A.14: Metrics on seed 1: a) life expectancy of agents, b) total number of resources present in the environment, c) size of the population and d) percentage of individuals with different amount of movement

At the very beginning, in the first phase A (fig A.14.A), the environment contains plenty of resources which leads to an increase in the population. In a second phase (fig A.14.B), the population seem to start to plateau while the amount of resources is still decreasing. This decrease in resources stops in phase C.

During phase A, B and C, the expectancy of the agents increases (fig A.14.a) suggesting that the agents are becoming better even though the environment is changing. The expectancy starts to plateau in phase D where it seems like the environment reaches a more or less stable state on some metrics.

Decrease in the amount of resources: A near tragedy-of-the-commons.

The decrease in the amount of resources in the environment at the beginning (fig A.14.b), seems to indicate that the evolving population as a whole depletes the resources in a greedy way even though more resources means a higher spawn of resources. The evolutionary path therefore seems to start by evolving a population which will go towards the tragedy of the common (which is here dampened by the fact that there are sparse spontaneous growth of resources). This is confirmed by looking at the environment after some time (fig A.16.a) where we can see that there are only few patch of resources in some corner of the map while the majority of the map is constantly depleted. This suggest that at least local tragedy of the common happens in our simulation.

Seed 2

Figure A.15 displays all metrics we discussed in Section A.3.3 for seed 1, this time measured for seed 2.

Diversity of eco-evolutionary path

We will now study some differences between seed 1 and 2.

In seed 1 sustainable and opportunistic travelers coexist during the whole evolution (fig A.14.D), while seed 2 has a majority of opportunistic travelers and some sparse period where low moving agents emerge (fig A.16.c). This may be explained by the differences in the environment led by the agents behavior. In fact seed 1 displays some area where there are big patches of resources (especially in the corner) (fig A.16.a)) and so where sustainable agents can easily take advantage of. On the other end in seed 2 (fig A.16.b), the map is completely depleted of patches of resources which only allows agents

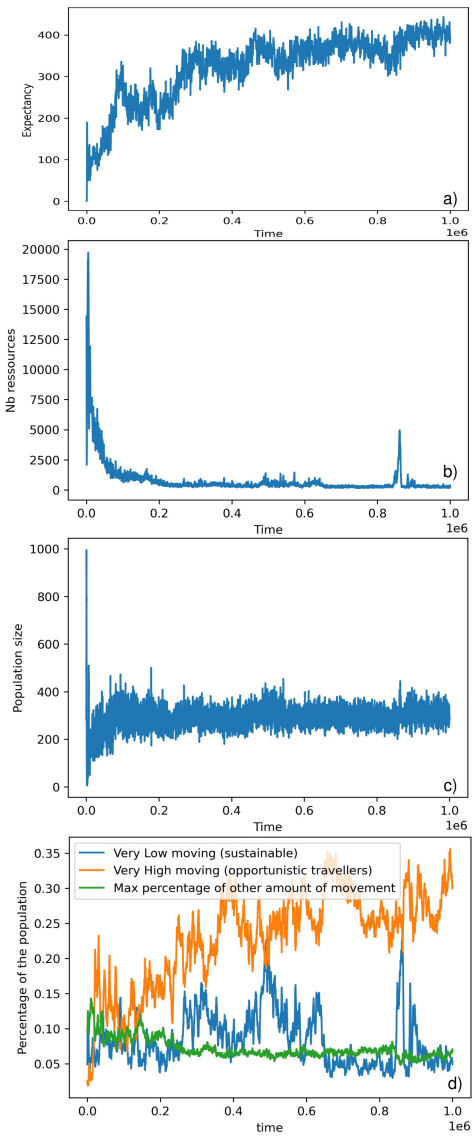


Figure A.15: Metrics on seed 2: a) life expectancy of agents, b) total number of resources present in the environment, c) size of the population and d) percentage of individuals with different amount of movement

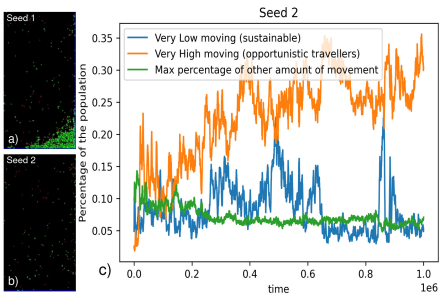


Figure A.16: Left: Diversity of environment between the 2 seeds (at timestep 600 000, zoom on the bottom right corner), here agents are in red; Right: Percentage of the population with different amounts of movement of seed 2

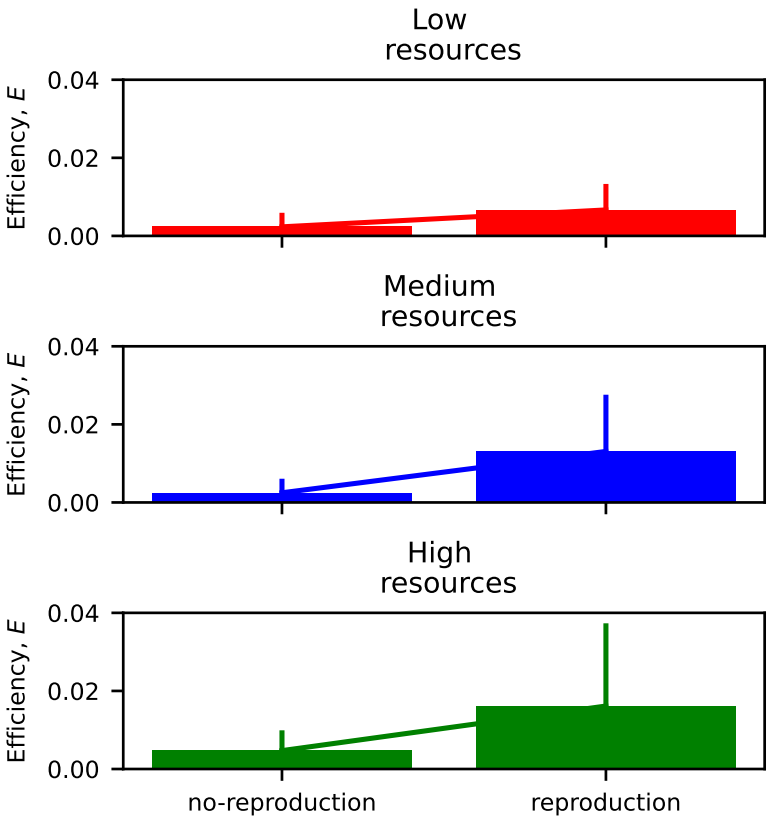


Figure A.17: Average efficiency across the population across different density levels with reproduction activated and deactivated. Activating reproduction leads to increased resource consumption.

to sustain on spontaneous regrowth on random spots of the map, which might explain why there are so much opportunistic travelers and nearly no sustainable behavior. The sustainable behavior pic we can see in seed 2 might be explained by timesteps where some spot of food were left for some time and so where bigger patches of resources emerged which might have favored some switches in behavior. See Videos 1.a and 1.b of the companion website for a better visualization of the dynamic and behavior of agents.

The (small) diversity of evolutionary and environment path between the 2 seeds we present are also an interesting feature of such eco-evo simulation.

Lab additional results

In this section, we provide additional results on the lab environment.

Fig A.17 shows the average efficiency averaged on the population on different density of resources (compared to only high resources task in fig 2.4.D) with reproduction activated and deactived, we observe that on every resource density, activating reproduction leads to increased resource consumption.

A.4 Appendix: emergence of agriculture

A.4.1 Details on the simulation.

The environment is a 30x30 grid, where channels represent : agent, seed green, green plant, seed yellow, yellow plant, water source, watered cell, purple plant.

Initial plants and water source location are randomly initialized.

During the last 10 timesteps before the end of the summer season, plants spread seeds in a 3x3 neighbourhood. The cell on which the plant is has a probability u_{color} to receive a grid at each of these timesteps while the other 8 cells have a probability $u_{color}/2$. u_{green} and u_{yellow} thus control the seed spreading of plants.

The plant that germinates is given by a categorical sampling over the classes (Nothing, yellow plant,green plant) with the probability vector $softmax(V)$ where $V = (S_{nothing}, S_{yellow\_sprout}, S_{green\_sprout})$ and

$$S_{green\_sprout} = \frac{N_{green\_seed}}{1 + N_{yellow\_seed} * \alpha_{compet}} * p_{green}$$

and

$$S_{yellow\_sprout} = \frac{N_{yellow\_seed}}{1 + N_{green\_seed} * \beta_{compet}} * p_{yellow}$$

and

$$S_{nothing} = clip(1 - (S_{green\_sprout} + S_{yellow\_sprout}), 0, 1)$$

p_{yellow} and p_{green} control the base probability of a seed to germinate while α_{compet} and β_{compet} control the competition between the two plants.

Seeds disappear from a cell with a probability of 0.0015 at each timestep and 0.15 at the beginning of summer if not sprouted.

Water on a soil cell evaporates with a probability of 0.005.

A.4.2 Agents architecture details.

The observation of the agent is a flattened vector of the local 11x11 grid around the agent, concatenated with the state of the inventory and the time of the season. The agent can see other agent in its 11x11 local information on the grid. We also concatenate the observation with the last action and reward.

Agents use a transformerXL based neural network using the implementation from Sec.4.3. The transformer has 2 attention layers with 4 heads, an embedding of size 128.

The agent takes as input the 128 last observations (memory size =128, effective extended memory as we use transformerXL with 2 layers is 256).

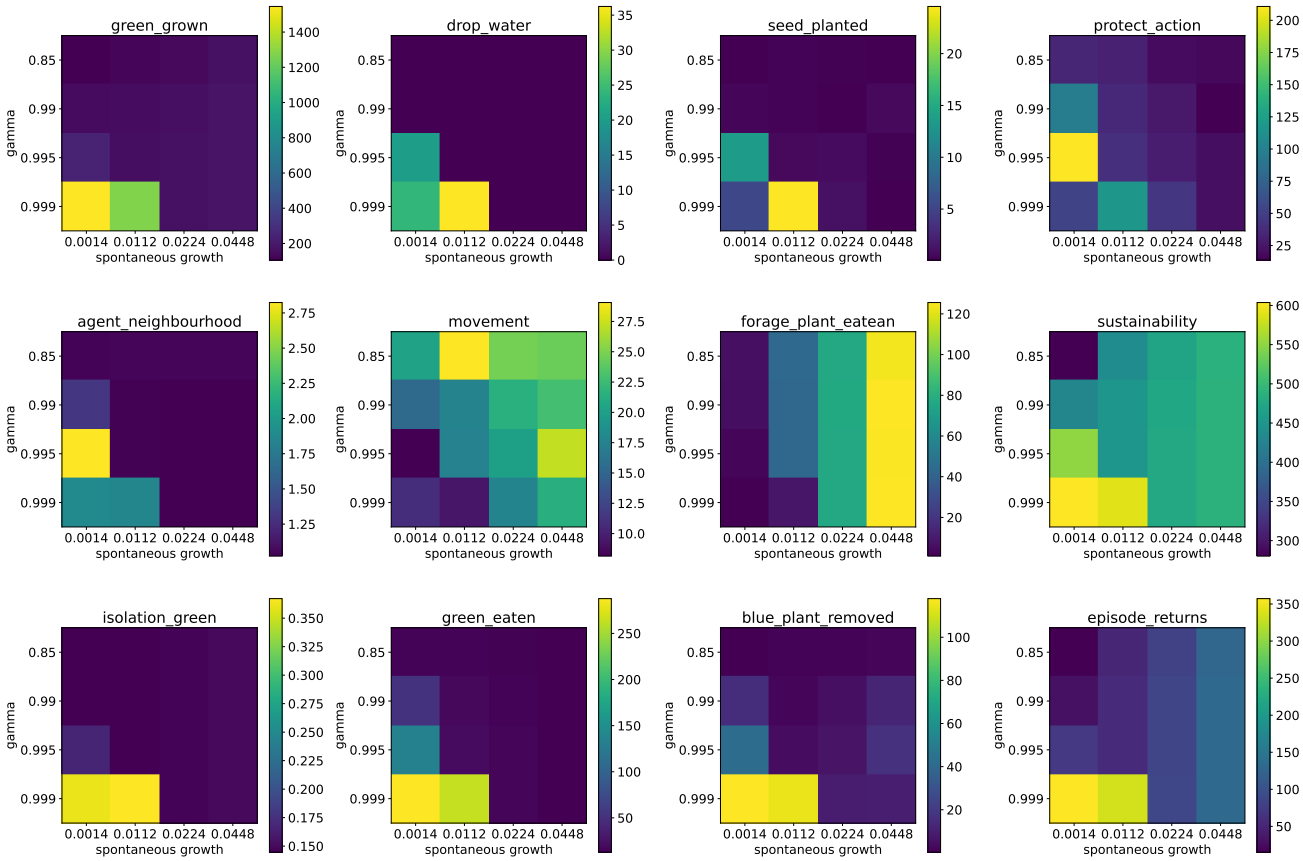


Figure A.18: Heatmap parameter analysis over the spontaneous growth probability of the purple plant p_{spont} and the discount factor (favoring exploration) γ. We observe a sharp transition from measures indicating foraging strategies (high p_{spont} and low λ_{gamma}; top right part) to measures indicating agricultural practices (low p_{spont} and high γ; bottom left part). Each parameter couple is tested over 3 seeds, we report the average value. The experiments were performed with $\lambda_{entr} = 0.036$

The agent policy has 3 actions head:

- 1st action head is to choose among the actions: move (up,down,left,right), pick, drop, protect.
- 2nd action head is to choose the type of resource (water, green seeds ,yellow seeds) to pick or drop when o
- 3rd action head allows to choose whether the agent will harvest on the cell he currently is. We made it a different head than the first one so that agent can move and harvest at the same timestep. (otherwise making the process of removing the competing green plants very costly in time, requiring long season to allow meaningful engineering, which is hard with RL because long credit assignment).

A.4.3 Additional results

We report in Fig.A.18, the parameter analysis heatmap (as in Fig.2.6) over gamma.

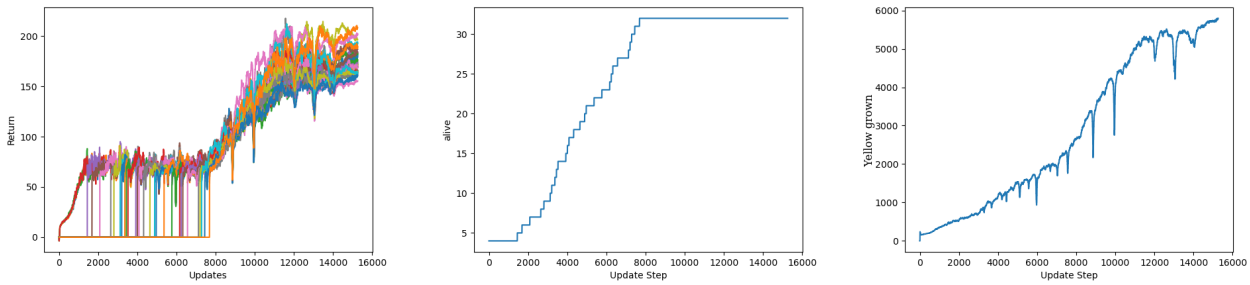


Figure A.19: Learning dynamic when agriculture emerges with reproduction activated. (left) total return vertical line represent new agent birth; (middle) number of agents; (right) Yellow plant grown. The number of agents steadily until reaching the max of 32, as new agents become additional workforce to grow more resources.

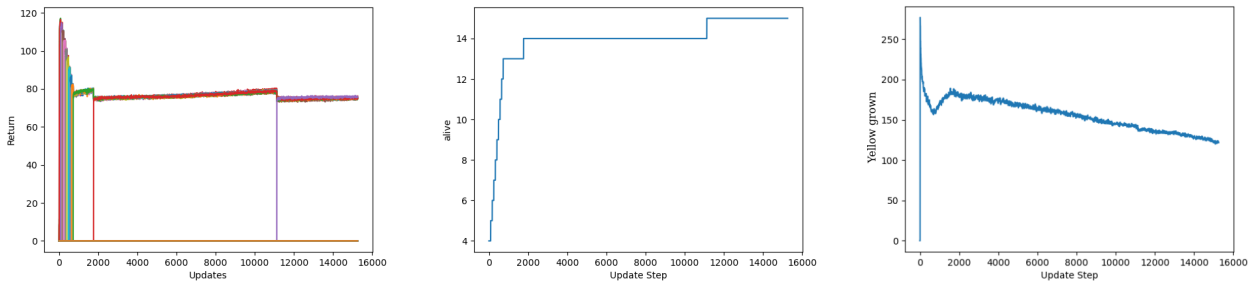


Figure A.20: Learning dynamic when agriculture does not emerges with reproduction activated. (left) total return vertical line represent new agent birth; (middle) number of agents; (right) Yellow plant grown. The number of agents reaches a plateau as the agents reach the capabilities of the environment.

Reproduction and population growth

In this section, we report preliminary results on experiments with population growth in the agricultural environment. We perform the same training as in Sec.2.2, except that we add reproduction and death during agents' training. More precisely, after a short warm-up period, we monitor the total return over an episode of every agent and :

- "Reproduce" the agent if it is above a certain threshold: we add an additional independent agent in the simulation initialized with the parameters of the agent that just reproduced.
- "Kill" the agent if it is below a certain threshold.

Due to computational limits, the maximum number of agents in these reported results is 32 (as we have to train all of them, requiring a lot of compute).

We report in Fig.A.19, the resulting learning dynamic in a simulation where agriculture emerged. We observe that the number of agents increases rapidly without any collapse. In particular, new agents seem to lead to a small decrease in reward, which is rapidly recovered, leading to new additional agents again. Interestingly, the agents produce much more resources (yellow plant) in the same environment as the experiments reported in Sec.2.8, showing the ability of the group of agents to eco-engineer the environment to produce more resources.

On the other hand, when agents do not discover agriculture, e.g. when p_{spont} is high, the population reproduces fast until it reaches a plateau (Fig.A.20). In fact, the foraging strategy does not benefit from additional agents; agents can't actively grow more resources than what the environment provides with p_{spont}. Therefore, the population of agents quickly reaches the capacity of the environment and can't improve its return.

A.5 Appendix: Emergence of Collective Open-Ended Exploration from Decentralized Meta-Reinforcement Learning

A.5.1 Forced Cooperation

We define a task tree as forced cooperative if at least one of the sampled subtasks requires both agents to solve it. We describe the three forced subtask types below:

Activate Landmarks: Now always two landmarks are randomly spawned at the edges of the environment. Agents are randomly assigned one of the two landmarks which they are able to activate. Additionally, the landmarks now have to be activated within ten environment steps of each other.

Meeting Point: One landmark is randomly spawned at the edges of the environment. The agents both have to be at the landmark and activate it within ten environment steps of each other.

Lemon Hunt: Now one agent is able to switch a specific object into the lemon object while the other agent is able to consume it.

A.6 Appendix: Autotelic Reinforcement Learning in Multi-Agent Environments

This appendix provides additional information about our set-up, implemenation details and results of section.3.2.

- Section A.6.1 describes our navigation tasks as MDPs;
- Section A.6.2 provides the hyper-parameters used in the main paper;
- Section A.6.3 intents to clarify how the different baselines we have evaluated differ algorithmically;
- Section A.6.4 contains an empirical analysis of how the complexity of our proposed algorithms changes with task difficulty;
- Section A.6.5 contains additional results. Specifically, Section A.6.5 shows the effect of environmental complexity, Section A.6.5 shows the effect of message size in the Goal-coordination game,

Section A.6.5 examines the usefulness of recurrent policies, Section A.6.5 replaces random sampling with Learning Progress, Section A.6.5 contains experiments where only cooperative goals are present in the environment, Section A.6.5 presents additional information about the "risky follower policy", Section A.6.5 contains specialization results and Sections A.6.5 and A.6.5 further analyze the baseline with both goals observable and CTDE.

A.6.1 Environment details

The environment is implemented in Python using Simple Playgrounds [350]. As a learning algorithm for the goal-conditioned policies we use RLlib's PPO implementation [438] and its multi-agent API with the PyTorch backend [439].

For the 3-landmarks environment the set of individual goals is $\{[001], [010], [100]\}$ and the set of cooperative goals is $\{[101], [011], [110]\}$. For the 6-landmarks environment the set of individual goals is $\{[000001], [000010], [000100], [001000], [010000], [100000]\}$ and the set of cooperative goals is $\{[110000], [101000], [100100],, [100010], [100001], [0110000], [010100], [010010], [010001], [001100], [001010], [001001], [000110], [000101], [000011], [100001]\}$

Observation space Agents are able to see each other and all the objects of the environment. We use object-centric representations, the observation vector contains the distance and the angle to each of the physical entities in the room (i.e walls, other agent, and landmarks). The order of the coordinates in the observation vector is preserved, e.g the first two coordinates are the distance and angle to the left wall. To make the navigation policy a goal-conditioned one, we concatenate the goal representation at the end of the observation vector to build the input to the networks. Observations are normalized between 0 and 1.

Action space We consider a discrete action space. Each agent is controlled by two actions: longitudinal force, and angular velocity. These actuators can take three different values: -1, 0, or 1.

Rewards and episodes Rewards are given independently to each agent conditioned on the agent's own goal. At each time step, if the goal is not fulfilled, the reward is 0, and 1 otherwise. All interactions with the environment are fully decentralized, each agent only has access to its own reward, and cannot see the reward of the others.

Once an agent gets a positive reward, the episode ends for them, i.e they cannot perform any other action but remain physically present in the environment. Episodes end either when both agents obtained their rewards or if a time limit is reached. At the beginning of an episode each agent is randomly placed inside the room, without touching any of the landmarks. The time limit in the environment was set to 250 and 500 time steps, for the 3 and 6 landmark instances respectively.

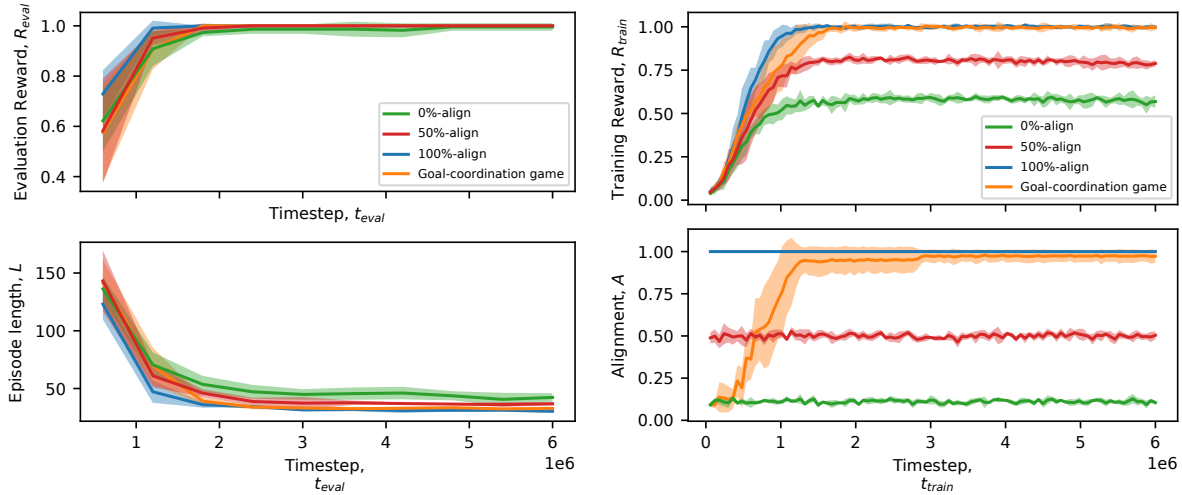


Figure A.21: Performance for the 3-landmarks environment during evaluation (left) and training (right) episodes for baselines exhibiting different levels of alignment and the Goal-coordination game

A.6.2 Hyperparameters

Hyper-parameters do not vary across methods.

PPO We base most of our design choices in the recommendations by [440]e:

- PPO policy loss with 0.3 clipping threshold.
- tanh as activation function for the networks. We don't use shared layers for the value and policy networks.
- Generalized Advantage Estimation (GAE) [441] with $\lambda = 0.9$
- A discount factor of $\gamma = 0.99$
- Adam optimizer [291] with a learning rate of 0.0003

From the many hyperparameters we can tune, we found that the batch size was the most relevant. After some test experiments, benchmarking results with the centralized uniform sampling baseline, we set this value to 16500 and 60000 time steps for the 3 and 6 landmarks experiments. We observed that usually a higher batch size was beneficial. For most of the hyperparameters we found that the defaults provided by RLlib were safe choices.

Goal-coordination game We use a softmax of temperature $T = \frac{1}{30}$ to sample messages m_l and goal g_f from the matrix. The update of the matrix is made with $\alpha = 0.1$ to dampen the changes of estimates of expected reward for each goal/message couple.

A.6.3 Illustration of baselines

In Figure A.22 we present an illustration of how the different methods we evaluate vary in terms of the information available to each agent and how it is used to condition its policy and value function.

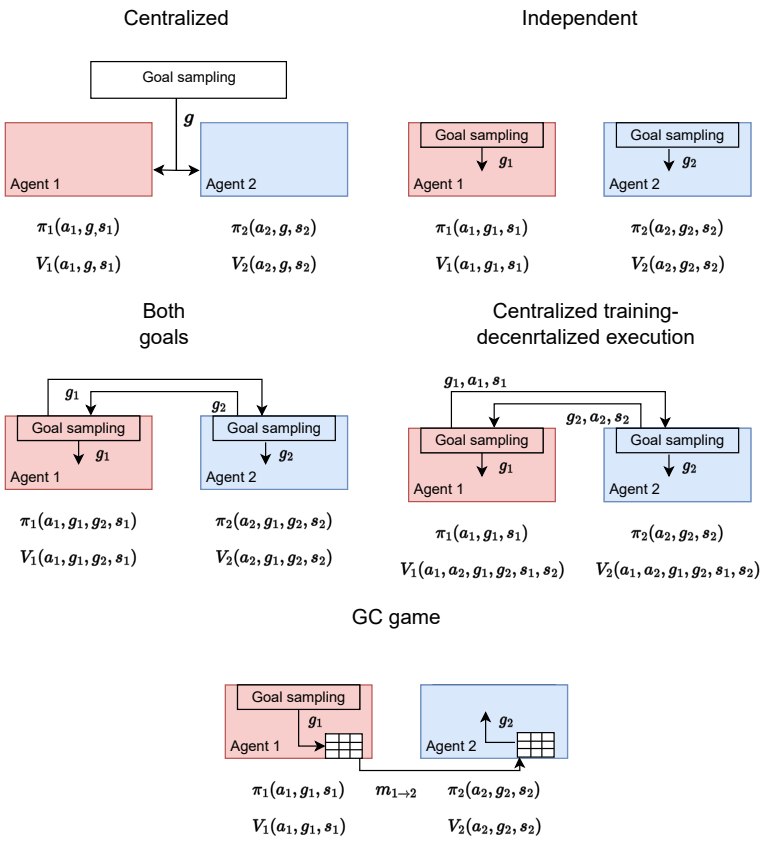


Figure A.22: Illustration of the different methods empirically evaluated in our work, where we indicate policies with π and value functions with V using the agent indexes as subscripts. The centralized baseline follows the algorithm proposed by [359], while the independent and both-goals baselines can be viewed as the equivalent of independent and joint learners proposed in the past [442] but for goals instead of state-actions.

A.6.4 Insights into training complexity

Understanding how our proposed solution performs as the number and difficulty of goals increases is useful for future applications of the Goal-coordination game to more complex settings. This algorithm is faced with the task of simultaneously learning how to align goals and learning how to solve them. In contrast, the centralized baseline (100%-aligned) is only faced with learning how to solve goals. To disentangle the difficulty of these two tasks we here study the complexity of these two methods in terms of two parameters: a) the size of the goal space b) the difficulty of achieving goals. To disentangle these two effects we make the following comparisons:

For a) we compare the training and evaluation performance of the two algorithms between the 3-landmarks and 6-landmarks environment when only cooperative goals are considered (thus only goals of equal difficulty). For b) we compare the training and evaluation performance between a setting where we train on both individual and cooperative goals and a setting with only cooperative goals, both in the 6-landmarks environment. Since the total number of goals is 21 and there are 6 individual goals, the latter setting contains about 70% more difficult goals.

We present results in Figure A.23 for the effect of goal space size and in Figure A.24 for the effect of goal difficulty. Regarding the goal space size, we observe that doubling the size of the goal space leads to about four times slower convergence. This is intuitive as the number of goals changes from 6 (in the 3-landmarks) to 21 (6-landmarks), so

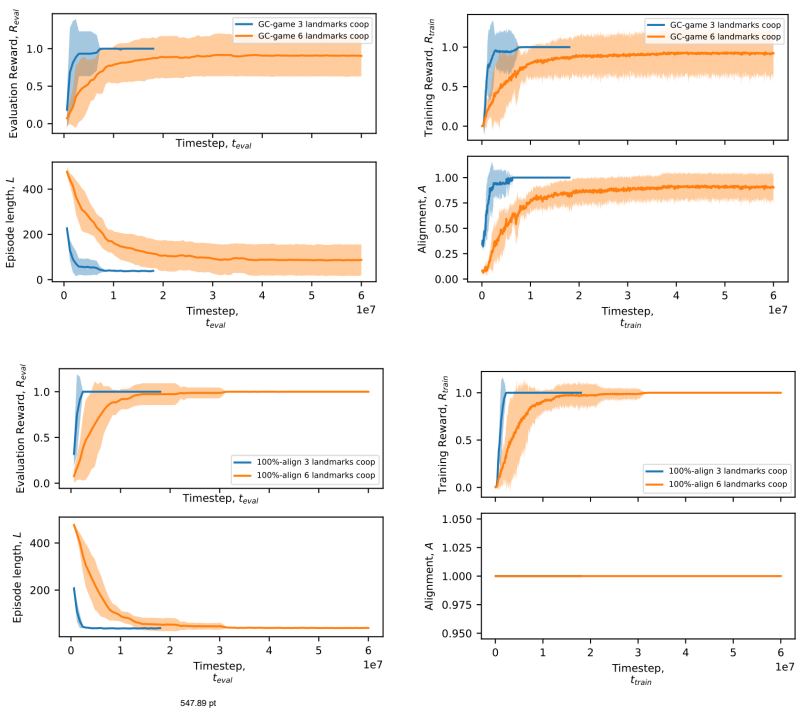


Figure A.23: Computational complexity of the Goal-coordination game and 100%-aligned for environments with different numbers of landmarks

more time is required to master the goals. This is also true for the centralized baseline, meaning that the increase in complexity is due to the need to learn more policies, rather than the need to align more goals. Regarding the goal difficulty, we see that the algorithm learns to solve quicker the task that has both individual and cooperative goals. Thus, although the number of goals increased the convergence time decreased. This is because the individual goals are solved more easily and then facilitate solving the cooperative goals. The same behavior is observed for the centralized baseline.

A.6.5 Additional results

3-landmarks environment

Figure A.21 contains the evaluation performance, on the left, and training performance, on the right for the 3-landmarks environment. We observe that, compared to the 6-landmarks environment, the population requires significantly less training time (about one order of magnitude smaller) and that differences across methods during evaluation are not as pronounced. During training, we observe that alignment is correlated with performance with the independent baselines collecting the least rewards. Thus, we conclude that our empirical conclusions generalize to simpler problem settings and that studying problems with increased task complexity is important for evaluating methods on the Dec-IMSAP.

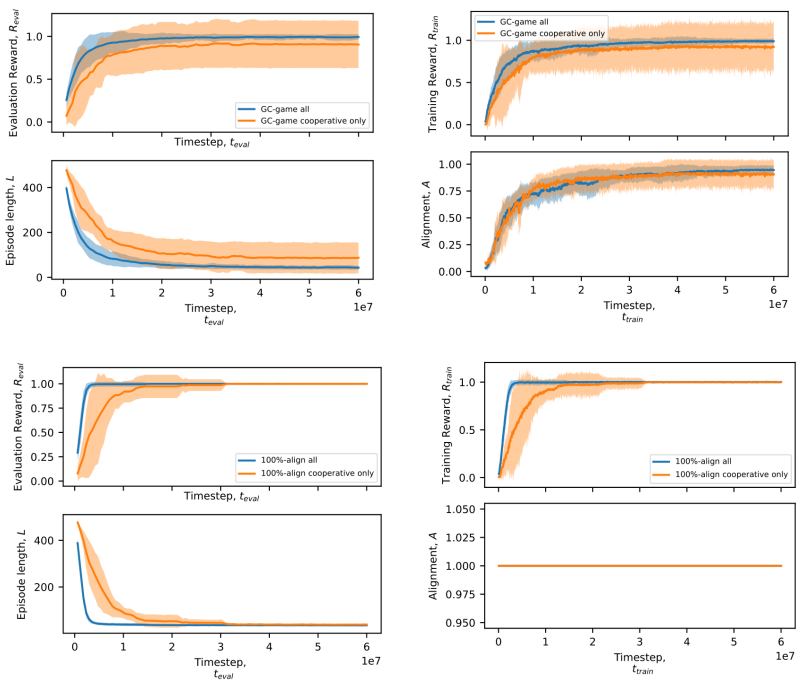


Figure A.24: Computational complexity of the Goal-coordination game and 100%-aligned for environments with goals of different difficulty.

Effect of message size

In Figure A.25 we study the effect of message size on the Goal-coordination game in the 6-landmarks environment by setting it to the smallest possible value ($M = 21$ is equal to the number of goals), a medium value ($M = 30$) and a high value ($M = 40$). We observe that evaluation performance does not vary significantly with message size except for the fact that small message size leads to slower convergence to the optimal policy. During training, we observe that small message size cannot reach perfect alignment and amasses slightly lower rewards. Thus, we conclude that the message size should be set to a value relatively higher than the number of goals but no further benefits are gained when it increases beyond that.

The main observation here is that, during training, using a message size equal to the number of goals (21) leads to sub-optimal alignment and rewards. By observing the matrix tables for this specific example, we understood that this is due to a deadlock: if one message-goal association is learned incorrectly early in training (where incorrectly means that the goal of the leader and follower are misaligned) then a column/row is reserved and cannot be used for the correct association. This leads to at least two goals being misaligned until the end of training. When we slightly increase the number of messages, on the other hand, a wrong association can be fixed later in training, because there are still enough degrees of freedom to align all messages.

Effect of scaling factor β

As we described in Section 3.2.7 the scaling factor β controls the relative importance of individual versus cooperative goals: increasing the value of β indicates a proportional decrease in the importance of

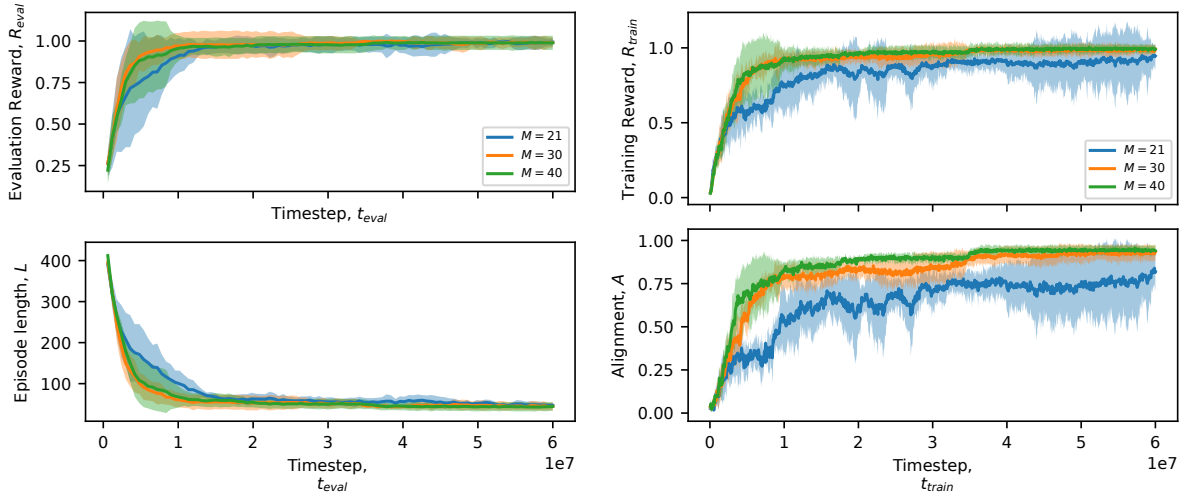


Figure A.25: Effect of message size M on the Goal-coordination game in the 6-landmarks environment during evaluation (left) and training (right) episodes

solving independent goals. To examine the effect of β we present the performance of the Goal-coordination game for different values ($\beta \in [1, 2, 4, 8]$) in Figure A.26. We observe that, for the Goal-coordination game, higher values of β lead to lower alignment: as cooperative goals are very rewarding in this case agents with the role of follower prefer them over individual ones even when the leader communicates about a cooperative goal. At the same time, low values of β lead to slower convergence to the optimal solution, as agents with the role of follower are not incentivized enough to choose cooperative goals, as they still receive rewards when they choose individual goals regardless of the leader's follower. Finally, contrasting the behavior of the Goal-coordination game to the other baselines in Figure A.26 shows that, by increasing β, the Goal-coordination game can amass more rewards during training that the centralized baseline. This is not surprising: as the agents learn this risky behavior or aligning cooperative with individual goals, they experience more rewarding episodes.

Recurrent policies

We have so far employed only feedforward policies in all our methods. We now study the effect of using a recurrent policy. Our intuition is that a recurrent policy can facilitate adaptation during the episode, as an agent can infer the direction the other is moving to and, perhaps, its goal. In Figure A.27 we compare this recurrent baseline with the other methods during evaluation and training trials. We observe that, during training, the recurrent policy with independent sampling (Recurrent 0% align) performs as badly as the independent feedforward baseline, while during evaluation, it is the worst-performing method. Thus, introducing a recurrent policy did not facilitate adaptation. Moreover, as the recurrent policy with centralized training (recurrent 100% align) converged to maximum reward, we can conclude that even with a recurrent policy the noisy training signal impacts learning when sampling goals independently. Since this method cannot lead to alignment, it is also negatively impacted by the large number

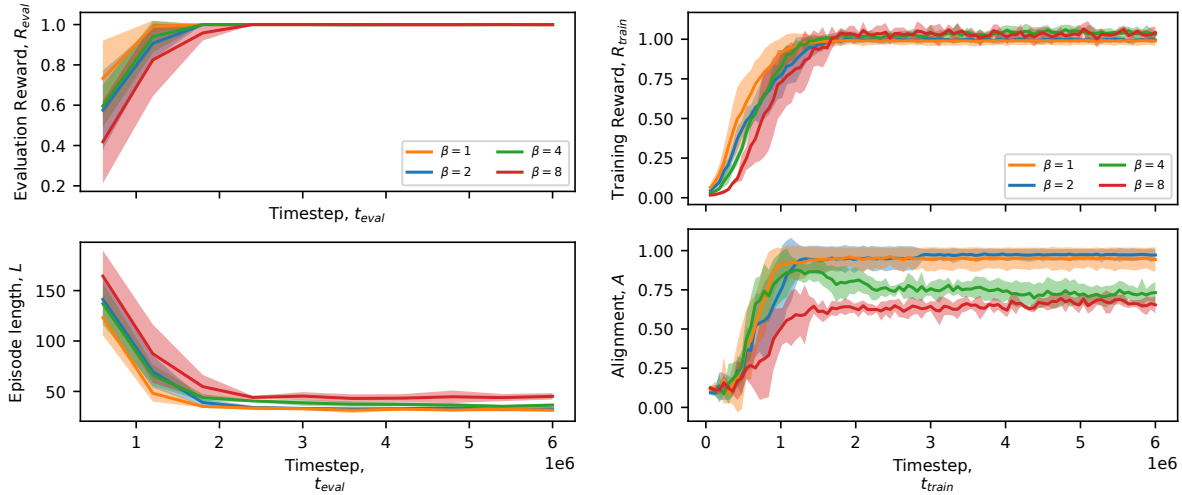


Figure A.26: Effect of scaling factor β on the Goal-coordination game in the 3-landmarks environment during evaluation (left) and training (right) episodes

of infeasible episodes. Finally, the fact that, with independent sampling, the recurrent policy performed worse than the feedforward one may mean that it may be even more sensitive to this noisy training signal. It is, however, possible that it may benefit from further hyperparameter tuning, for example an increase in the size of the neural network.

Learning progress for sampling goals

Throughout the manuscript we have considered that, when an agent sets its own goal, it does so by randomly sampling within the goal space. This is the simplest form of intrinsic motivation. An interesting question is how our independent baseline would behave if the sampling of goals is performed in a more sophisticated way, for example based on the competence of an agent. Learning progress is such a type of intrinsic motivation that has been previously employed in single-agent settings [250, 407]. Here, we extend learning progress to our two-agent setting. Our main motivation for this small study is to test whether introducing learning progress will indirectly lead to goal alignment. Our intuition is that, by helping the agent focus on easier tasks first and then tackle the more challenging ones, this approach may lead to a curriculum from independent (easy) to cooperative (difficult) goals and that this may facilitate alignment.

At the start of an episode, each agent has a vector $LP \in [-1, 1]^K$. Each coordinate of this vector is an approximation of the derivative in time of the competence of that agent for solving each goal. Goals are selected using a $\epsilon -$ greedy strategy and a proportional probability matching using the absolute value of the LP. For each agent i, the probability of selecting goal g_i is given by:

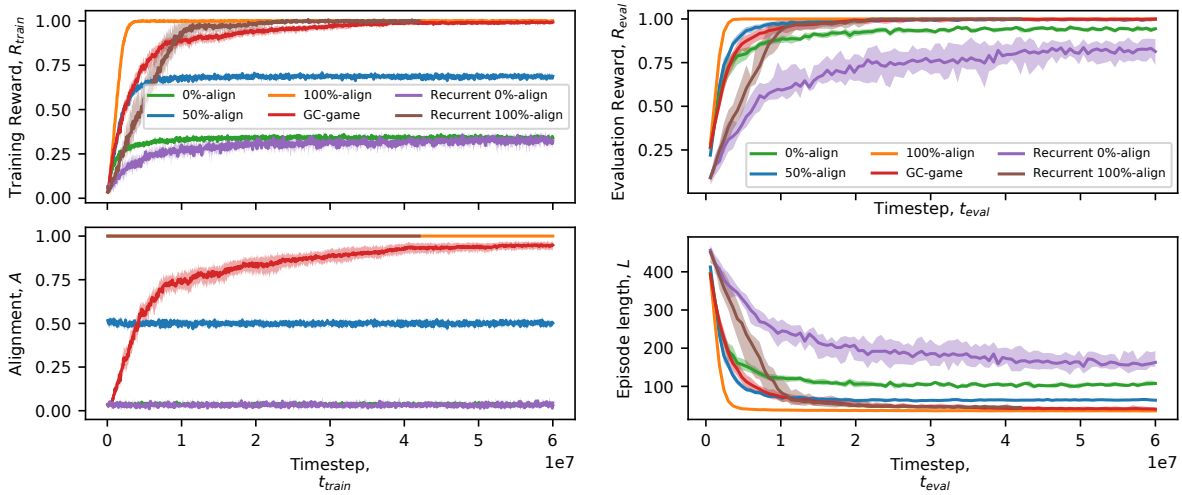


Figure A.27: Comparison of the recurrent policy to the other baselines

$$p(g_i) = \epsilon \times \frac{1}{K} + (1-\epsilon) \times \frac{|LP^a_{g_i}|}{\sum_{j=1}^{K} |LP^a_{g_i}|}$$

The use of the absolute value makes agents concentrate both in goals that are currently being learned or forgotten. In the original implementation, LP is computed during evaluation rounds which provide a better signal than training data. However, in our multi-agent context, goal selection should be decentralized. Therefore, the first change we make to the strategy is to get rid of the evaluation rounds, and only estimate the learning progress based on experience from training. As we also want to work in a fully-decentralized context, at goal selection time we won't assume that one agent can have privileged information from the other (e.g access to other agent's goal). Each agent keeps a competence vector $C[n] \in [0,1]^K$ whose entries $C[n]_i$ is the moving average of the rewards obtained when the agent selected goal i for the n time during training. This average is defined by a window length w, which is the number of episodes we want to include for computing it. Then, the learning progress at time n is:

$$LP[n]_i = C[n]_i - C[n-w]_i$$

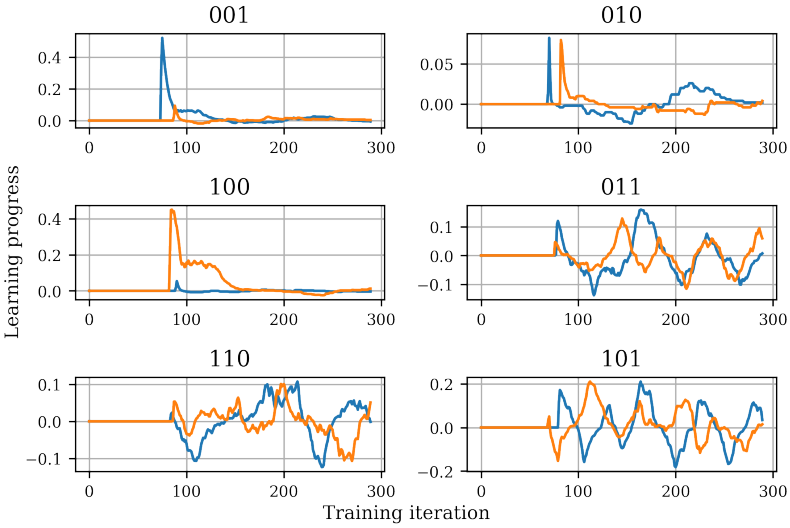


Figure A.28: Learning progress estimate for each goal and two agents (blue and orange).

Our experiments showed that estimating LP did not improve the performance of the independent baseline. As we observe in Figure A.28 LP values are very noisy and therefore lead to sampling goals relatively randomly, certainly not showing a curriculum from individual to cooperative goals. This is not surprising: estimating LP in multi-agent environments is challenging, because the competence of one agent depends on the competence of other agents as well. One agent's LP and competence in a cooperative goal doesn't only depend on the behavior of that agent, but also on the rest of them. Furthermore, this estimate includes data from episodes where the pair of goals was impossible to solve (e.g one agent sampled one cooperative goal and the other an individual goal that cannot be solved at the same time). This is particularly evident for cooperative goals: for individual goals the LP plot look similar in shape to the ones presented in a previous work for a single-agent setting [250], while for cooperative goals the curves are too noisy and do not converge.

Cooperative goals only

In this experiment, individual goals are removed both in training and evaluation. This means that the leader can only sample cooperative goals and the follower can only choose cooperative goals from its matrix. We observe the same conclusion as in the experiments with all goals but with bigger gap between methods both for the 3 landmarks Fig.A.29 and the 6 landmarks Fig.A.30 cases. Also in this setup we see that the Goal-coordination game converges to 100% alignment during training and converges to the same performances as the 100% alignment method both in term of reward and episode length.

Illustration of the "risky follower"

We have described the "risky follower" behavior in Section 3.2.7, where we defined it as a matching between a leader's individual goal and a follower's cooperative goal and presented the communication matrix

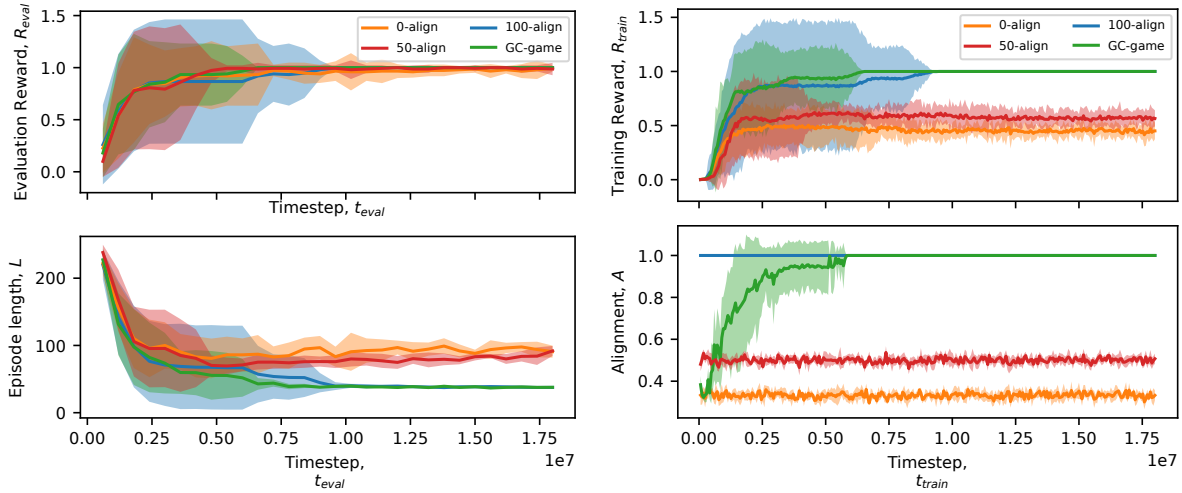


Figure A.29: Performance for the 3-landmarks environment with only cooperative goals during evaluation (left) and training (right) episodes for baselines exhibiting different levels of alignment and the Goal-coordination game

that leads to it for the 6-landmarks environment. We now illustrate it for the 3-landmarks environment (with $\beta = 4$) in Figure A.31.

We can even see on the training reward in Figure A.32 that the risky follower behavior is used as we can see that the average reward of the goal coordination game is higher than the theoretical maximum of centralized training. In fact, in the case of the centralized training, the maximum average reward for one agent is capped by $P(sampling\_individual\_goal)*R(individual\_goal\_fulfilled)+P(sampling\_cooperative\_goal)*R(cooperative\_goal\_fulfilled) = 0.5*0.25+0.5*1 = 0.625$, and so the average reward of the sum of the 2 agents is capped by 1.25. While on the other hand, the Goal-coordination game can allow the follower agent to get more reward when an individual goal is sampled by the leader and so have a higher maximum average sum of reward.

Specialization

We defined specialization as the ratio of the episodes in which the agent went to its preferred landmark when following a cooperative goal in Section 3.2.7. We now visualize the values of specialization we reported on the left of Figure A.33.

Both goals observable

The objective of this experiment is to see if : a) agents that observe both goals can learn to ignore infeasible episodes b) agents that learn to ignore infeasible episodes can achieve the same performance with the centralized baseline. This will help us understand if the independent baseline fails due to the noisy updates caused by infeasible episodes. As in our experiments the agent does not know the goal of the other agent, it might not know if the fail was due to it's policy or simply that the goal of the other was incompatible. By giving the goal of the other agents, we expect agents to learn to discard episode

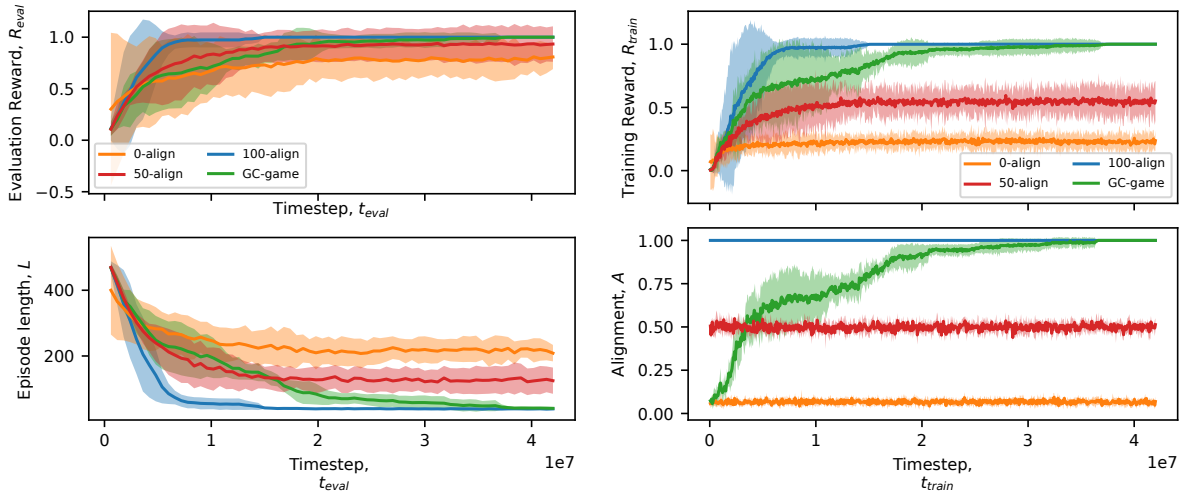


Figure A.30: Performance for the 6-landmarks environment with only cooperative goals during evaluation (left) and training (right) episodes for baselines exhibiting different levels of alignment and the Goal-coordination game

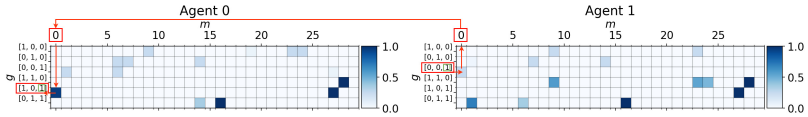


Figure A.31: Example of the "Risky follower behavior" in the 3 landmarks case and reward multiplier $\beta = 4$. Leader is agent 1 which samples goal [0,0,1] and send message 0. Agent 1 is the follower and interprets message 0 as [1,0,1] which is cooperative and compatible with goal of agent 0.

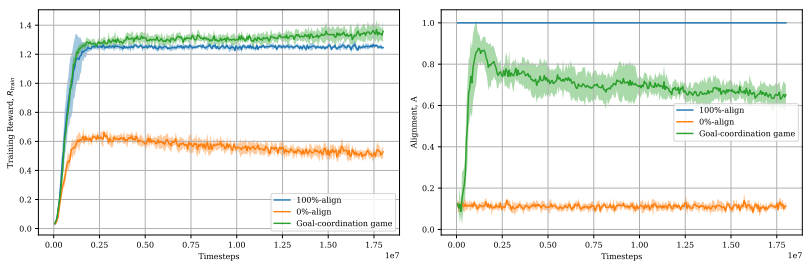


Figure A.32: Training performances in the 3 landmarks case and $\beta = 4$, we can see on left that goal coordination game exceeds the performances of the centralized (which attains its max value)

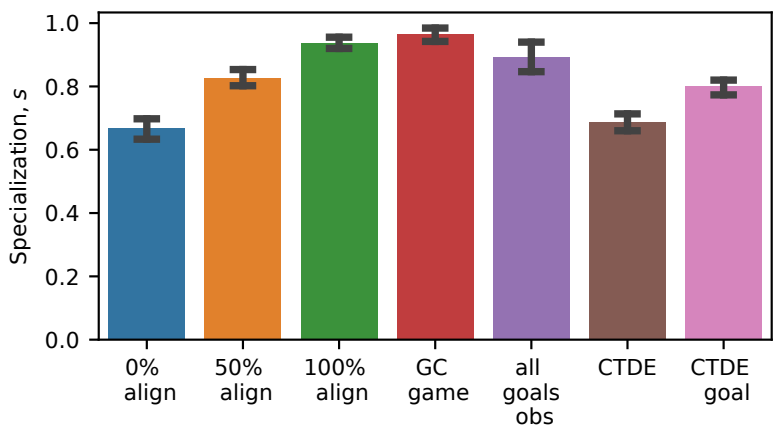


Figure A.33: Specialization for the 6-landmark environment with $\beta = 4$

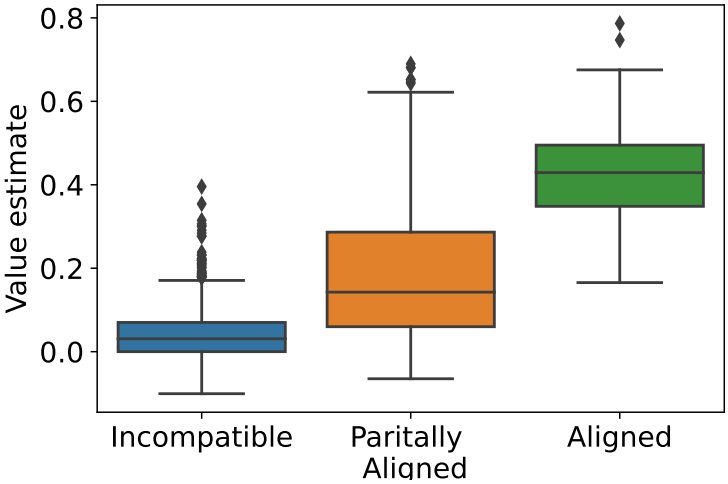


Figure A.34: Learned value functions of the "both goal in the obs" baseline applied to different types of couple of goals. The boxplots distributions take into account several couple of goal and several seeds

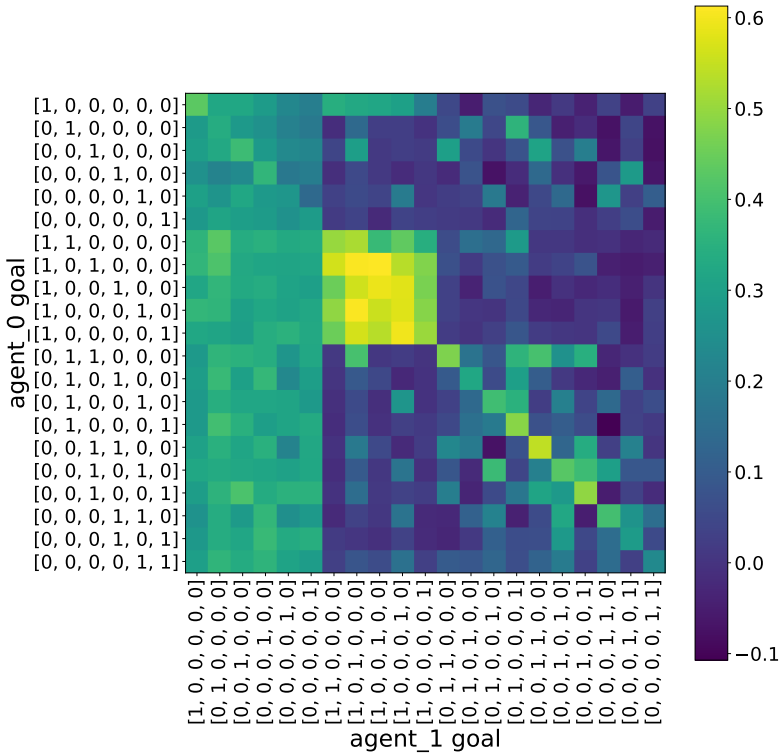


Figure A.35: Value function of agent 1 (in one seed), in the both goals in the obs baseline, applied to different couples of goals for agent 0 and agent 1.

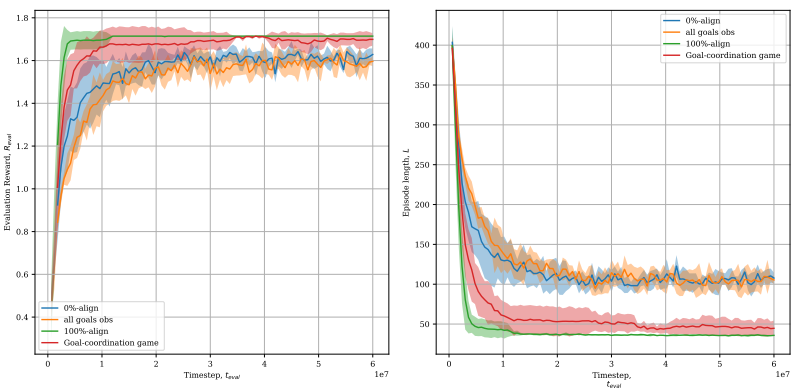


Figure A.36: Comparing the performances of the all goals in the obs in evaluation, in the 6 landmarks, $\beta = 2$ case

where goals are incompatible by giving them low value (expecting no reward).

In this section we thus study the case where every agent has access to both its goal and the goal of the other agent in the observation given to the policy and value network. We study this while being in the independent sampling case: agents sample their goal independently of the other at the beginning of the episode.

In this section, we separate couples of cooperative goals into 3 categories: 1) incompatible goals are goals where there is no overlap at all, meaning that there is no common landmark in the cooperative goals of both agents (eg [1,1,0,0,0,0] and [0,0,1,1,0,0]; 2) Partially-aligned goals are cooperative goals which overlap on 1 of the landmark (eg [1,1,0,0,0,0] and [0,1,1,0,0,0]); 3) aligned goals are when goals are the same.

When looking at the distribution of value (given by the trained value function) on the different categories of couple of cooperative goals (incompatible, partially aligned and aligned) across 5 seeds in Figure A.34 , we can clearly see that the training learned to give very low value to incompatible goals and even to some of the partially compatible goals)

Even though training seems to learn that some goal associations are incompatible, and even though agents seem to specialize in Figure A.33 the performances of the "Both goals observable" is still no better than independent without access to the goal of the other agent in Figure A.36.

The fact that some values are high for the partially aligned goals case in Figure A.34 is due to the specialization. If your goals overlap on only one landmark and you know that the other agent specialized to go to this common landmark when he has this goal, then you can go to your other landmark to get the reward. For example, looking at the matrix of agent 1 in one of the seed in Figure A.35, we can see that when agent 0 has the goal [1,0,1,0,0,0], agent 1 has a high value for goals [1,1,0,0,0,0],[1,0,1,0,0,0],[1,0,0,1,0,0],[1,0,0,0,1,0],[1,0,0,0,0,1] which all contain landmark 1. This seems to indicate that agent 0 has a bias toward landmark 1 when having goal [1,0,1,0,0,0], which is exploited by agent 1. For example if agent 0 has goal [1,0,1,0,0,0] and agent 1 has goal [1,1,0,0,0,0], agent 1 will go to the 2nd landmark as he learned that agent 0 will go to the first one when he has goal [1,0,1,0,0,0].

Analysis of CTDE

As we saw in the main paper in Section 3.2.7 in our discussion of Figure 3.14, the CTDE baseline performed better than the both-goals condition, which exhibited performance as bad as the independent baseline. As we show in Figure A.38, CTDE has, similarly to the both-goals baseline learned how to detect infeasible episodes. Why does this method outperform both-goals, then? As we explained in our description of methods in Figure A.22, the two methods differ in two respects: a) the policies in both-goals are conditioned on both goals while only on an individual's goal for the CTDE b) the value function has access

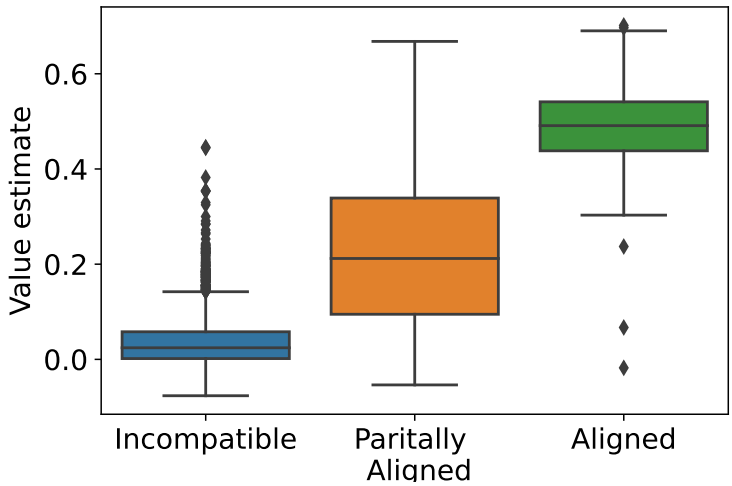


Figure A.37: Learned value functions of the CTDE baseline applied to different types of couple of goals. The boxplots distributions take into account several couple of goal and several seeds

to, in addition to both goals, the states and actions of both agents. To disentangle the effect of these two differences, we evaluate an additional method: a CTDE whose value function is conditioned on both goals but on an individual's state and action. We compare the performance of these three slightly different methods in Figure A.39. We observe that the new variant performs as well as CTDE, which indicates that the worse performance of both-goals is due to its conditioning on both goals. This could be due to this method requiring further tuning, due to the increase in the size of the learning space, or due to the higher difficulty of learning policies conditioned on both goals.

To further understand the behavior of agents under CTDE, we also analyze the specialization of these variants and contrast it to the specialization of all baselines in Figure A.33, where we observe that CTDE has lower specialization. In videos collected during evaluation trials, we also observed that CTDE agents exhibit intra-episode adaptation. This indicates that CTDE is a promising approach towards unsupervised skill acquisition in our multi-agent settings. Yet, it has not reached the performance of our proposed algorithm, although it introduces a need for centralization. This is arguably due to the fact that this method is still plagued by the presence of infeasible episodes.

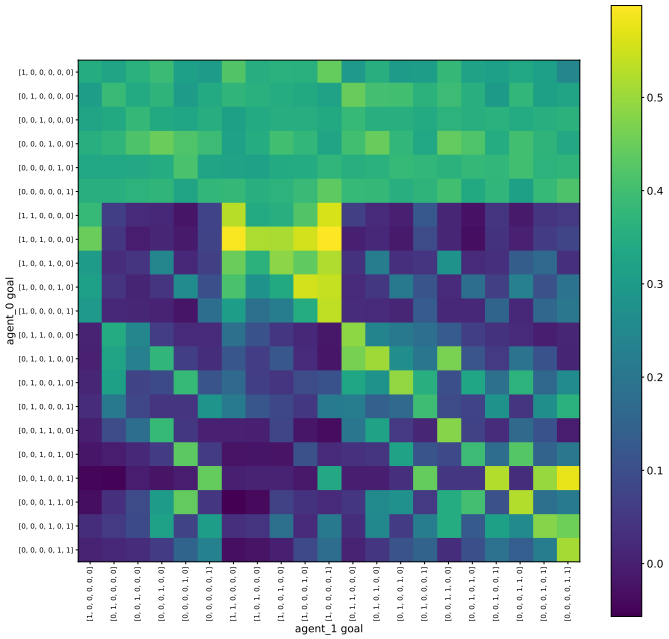


Figure A.38: Value function of agent 0 (in one seed), in the CTDE baseline, applied to different couple of goals for agent 0 and agent 1

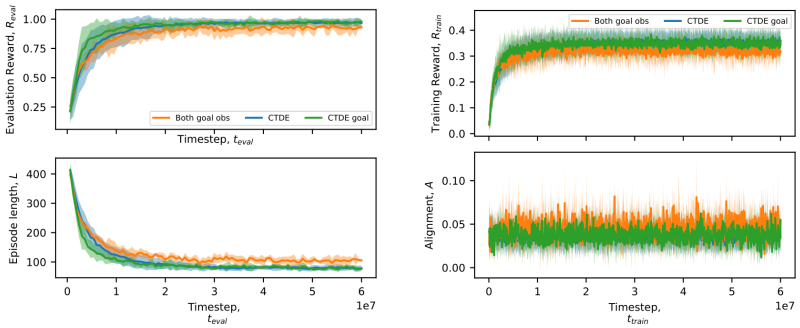


Figure A.39: Comparing CTDE and two variants: the drop in performance happens when we condition the policies on both goals.

A.7 Appendix: TransformerXL results on craftax

We report in Fig.A.40 and Fig.A.41 the achievement's success rate along training of our transformerXL-based agents trained with PPO presented in Sec.4.3. The instructions to reproduce these results are available in the code repository.

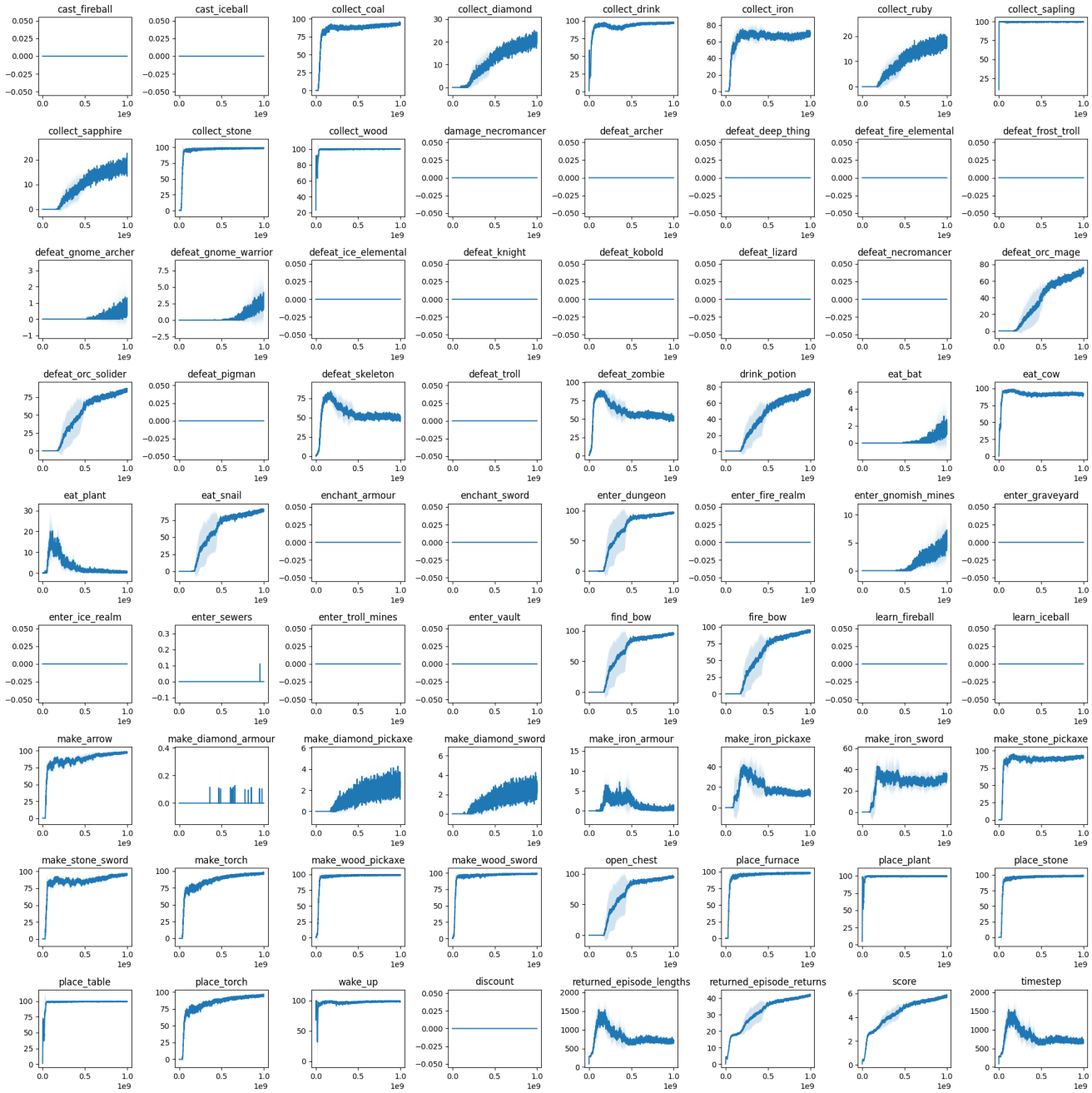


Figure A.40: Achievements succcess rate on craftax [232] of our implementation of tranformerXL base agents trained with PPO (Sec.4.3) over 1e9 timesteps.

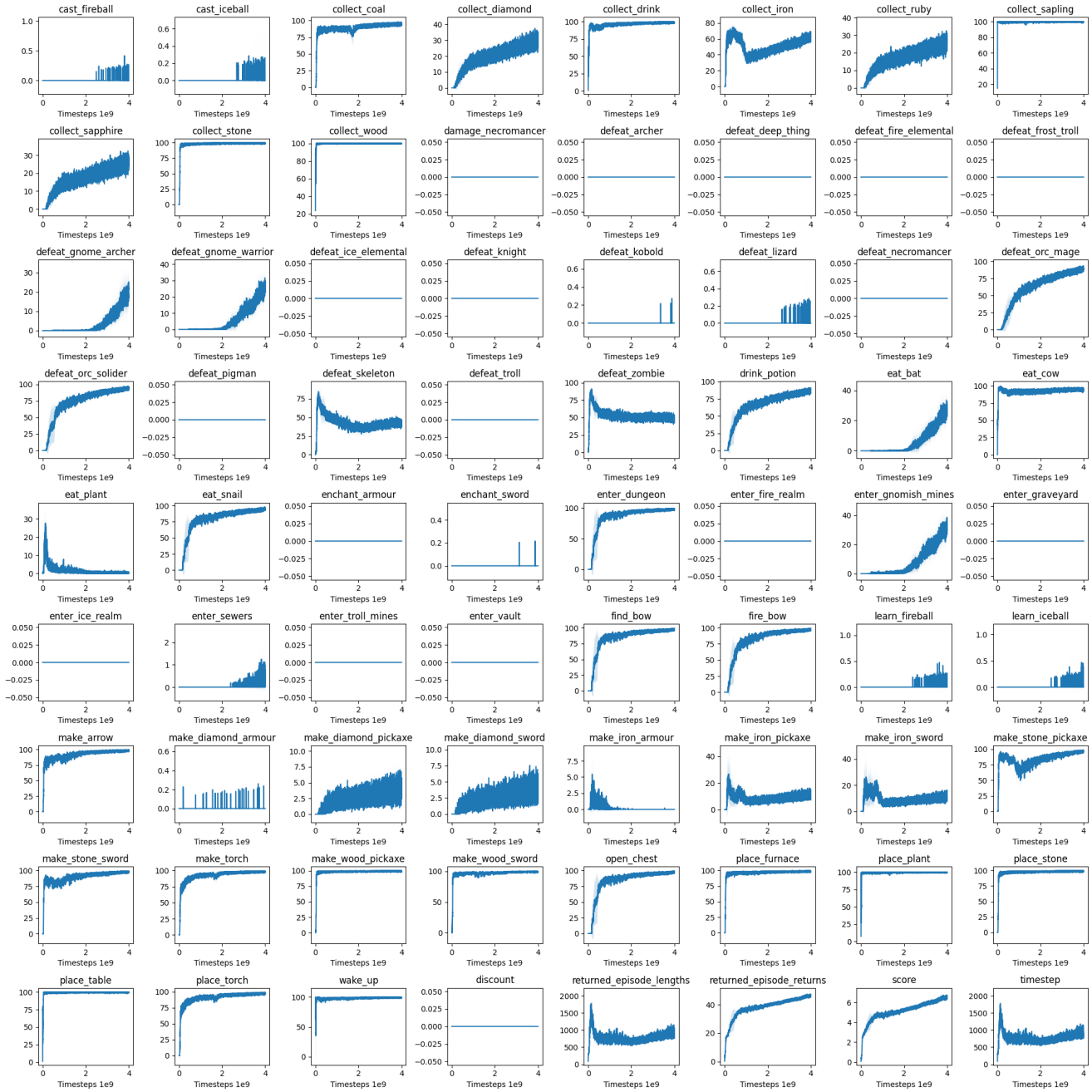


Figure A.41: Achievements succcess rate on craftax [232] of our implementation of tranformerXL base agents trained with PPO (Sec.4.3) over 4e9 timesteps.

Bibliography

Here are the references in citation order.

[1] Kenneth O Stanley, Joel Lehman, and Lisa Soros. "Open-endedness: The last grand challenge you've never heard of." In: *O'Reilly Online* (2017) (cited on pages 1, 4, 49).

[2] Mayalen Etcheverry. "Curiosity-driven AI for Science: Automated Discovery of Self-Organized Structures." PhD thesis. Université de Bordeaux, 2023 (cited on pages 1, 158).

[3] Jeff Clune. "AI-GAs: AI-generating algorithms, an alternate paradigm for producing general artificial intelligence." In: *arXiv:1905.10985 [cs]* (Jan. 2020). arXiv: 1905.10985. (Visited on 06/17/2020) (cited on pages 1, 3, 4, 11, 148, 156).

[4] Alan Dorin and Susan Stepney. "What Is Artificial Life Today, and Where Should It Go?" In: *Artificial Life* 30.1 (Feb. 2024), pp. 1–15. DOI: 10.1162/artl_e_00435 (cited on page 1).

[5] Lisa Soros and Kenneth Stanley. "Identifying necessary conditions for open-ended evolution through the artificial life world of chromaria." In: *Artificial Life Conference Proceedings*. MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info ... 2014, pp. 793–800 (cited on pages 1, 3, 4, 8, 13, 16, 61).

[6] Mark A Bedau et al. "Open problems in artificial life." In: *Artificial life* 6.4 (2000), pp. 363–376 (cited on pages 1, 7, 8, 15).

[7] Carlos Gershenson et al. "Self-organization and artificial life." In: *Artificial Life* 26.3 (2020), pp. 391–408 (cited on pages 2, 16).

[8] Bert Wang-Chak Chan. *Lenia - Biology of Artificial Life*. 2019. URL: https://arxiv.org/abs/1812.05433 (cited on pages 2, 6, 24, 27, 32, 33, 35, 38, 41, 47, 48, 54, 161, 162, 170, 182).

[9] Bert Wang-Chak Chan. "Lenia and Expanded Universe." In: *The 2020 Conference on Artificial Life*. 2020, pp. 221–229. DOI: 10.1162/isal_a_00297 (cited on pages 2, 6, 24, 27, 32, 33, 35, 38, 41, 47, 48, 54, 132, 161, 162, 170, 182).

[10] Kenneth O. Stanley and Joel Lehman. *Why Greatness Cannot Be Planned: The Myth of the Objective*. Springer Publishing Company, Incorporated, 2015 (cited on pages 2, 3, 156).

[11] Andrew N Sloss and Steven Gustafson. "2019 evolutionary algorithms review." In: *arXiv preprint arXiv:1906.08870* (2019) (cited on pages 2, 8).

[12] Pradnya A Vikhar. "Evolutionary algorithms: A critical review and its future prospects." In: *2016 International conference on global trends in signal processing, information computing and communication (ICGTSPICC)*. IEEE. 2016, pp. 261–265 (cited on pages 2, 8).

[13] Yann LeCun. "A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27." In: (2022) (cited on page 2).

[14] George Cybenko. "Approximation by superpositions of a sigmoidal function." In: *Mathematics of control, signals and systems* 2.4 (1989), pp. 303–314 (cited on page 2).

[15] Yann LeCun et al. "Gradient-based learning applied to document recognition." In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324 (cited on pages 2, 3).

[16] Alex Sherstinsky. "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network." In: *Physica D: Nonlinear Phenomena* 404 (2020), p. 132306 (cited on page 2).

[17] Ibomoiye Domor Mienye, Theo G Swart, and George Obaido. "Recurrent neural networks: A comprehensive review of architectures, variants, and applications." In: *Information* 15.9 (2024), p. 517 (cited on page 2).

[18] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory." In: *Neural Comput.* 9.8 (Nov. 1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735 (cited on pages 2, 110).

[19] A Vaswani. "Attention is all you need." In: *Advances in Neural Information Processing Systems* (2017) (cited on page 2).

[20] Albert Gu, Karan Goel, and Christopher Ré. "Efficiently modeling long sequences with structured state spaces." In: *arXiv preprint arXiv:2111.00396* (2021) (cited on page 2).

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks." In: *Advances in neural information processing systems* 25 (2012) (cited on pages 2, 3).

[22] Mathilde Caron et al. "Emerging properties in self-supervised vision transformers." In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 9650–9660 (cited on page 2).

[23] I Sutskever. "Sequence to Sequence Learning with Neural Networks." In: *arXiv preprint arXiv:1409.3215* (2014) (cited on page 2).

[24] Jacob Devlin. "Bert: Pre-training of deep bidirectional transformers for language understanding." In: *arXiv preprint arXiv:1810.04805* (2018) (cited on page 2).

[25] R Thomas McCoy et al. "Embers of autoregression show how large language models are shaped by the problem they are trained to solve." In: *Proceedings of the National Academy of Sciences* 121.41 (2024), e2322420121 (cited on page 2).

[26] Iman Mirzadeh et al. "Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models." In: *arXiv preprint arXiv:2410.05229* (2024) (cited on page 2).

[27] Junbing Yan et al. "Do Large Language Models Understand Logic or Just Mimick Context?" In: *arXiv preprint arXiv:2402.12091* (2024) (cited on page 2).

[28] Laura Ruis et al. "Procedural Knowledge in Pretraining Drives Reasoning in Large Language Models." In: *arXiv preprint arXiv:2411.12580* (2024) (cited on page 2).

[29] Edwin Zhang et al. "Transcendence: Generative Models Can Outperform The Experts That Train Them." In: *arXiv preprint arXiv:2406.11741* (2024) (cited on page 2).

[30] Richard Sutton. *The Bitter Lesson*. URL: http://www.incompleteideas.net/IncIdeas/BitterLesson.html (visited on 12/11/2024) (cited on page 2).

[31] Joel Lehman and Kenneth O. Stanley. "Novelty Search and the Problem with Objectives." In: *Genetic Programming Theory and Practice IX*. Ed. by Rick Riolo, Ekaterina Vladislavleva, and Jason H. Moore. Genetic and Evolutionary Computation. New York, NY: Springer, 2011, pp. 37–56. DOI: 10.1007/978-1-4614-1770-5_3. (Visited on 10/21/2022) (cited on pages 3, 52).

[32] Thomas S Ray. "An approach to the synthesis of life." In: *The philosophy of artificial life* (1996), pp. 111–145 (cited on pages 3, 8, 13, 16).

[33] Thomas Miconi. "Evosphere: evolutionary dynamics in a population of fighting virtual creatures." In: *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*. IEEE. 2008, pp. 3066–3073 (cited on pages 3, 8, 13, 16).

[34] Larry Yaeger et al. "Computational genetics, physiology, metabolism, neural systems, learning, vision, and behavior or Poly World: Life in a new context." In: *SANTA FE INSTITUTE STUDIES IN THE SCIENCES OF COMPLEXITY-PROCEEDINGS VOLUME-*. Vol. 17. ADDISON-WESLEY PUBLISHING CO. 1994, pp. 263–263 (cited on pages 3, 4, 8, 13, 16).

[35] Alastair Channon et al. "Improving and still passing the ALife test: Component-normalised activity statistics classify evolution in Geb as unbounded." In: *Artificial Life* 8 (2003), pp. 173–181 (cited on pages 3, 8, 13, 16, 157).

[36] Alastair Channon. "Unbounded evolutionary dynamics in a system of agents that actively process and transform their environment." In: *Genetic Programming and Evolvable Machines* 7 (2006), pp. 253–281 (cited on pages 3, 8, 13, 16, 157).

[37] Richard E Lenski et al. "The evolutionary origin of complex features." In: *Nature* 423.6936 (2003), pp. 139–144 (cited on pages 3, 8, 13, 16).

[38] Lee Spector, Jon Klein, and Mark Feinstein. "Division blocks and the open-ended evolution of development, form, and behavior." In: *Proceedings of the 9th annual conference on genetic and evolutionary computation*. 2007, pp. 316–323 (cited on pages 3, 6, 8, 13, 16).

[39] Bowen Baker et al. "Emergent Tool Use From Multi-Agent Autocurricula." In: *International Conference on Learning Representations*. 2020 (cited on pages 3, 16, 31, 49, 106, 132).

[40] Jimmy Secretan et al. "Picbreeder: evolving pictures collaboratively online." In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2008, pp. 1759–1768 (cited on page 3).

[41] Clément Moulin-Frier. "The Ecology of Open-Ended Skill Acquisition." Habilitation à diriger des recherches. Université de Bordeaux (UB), Dec. 2022 (cited on pages 3, 70).

[42] Nicolas Bredeche and Jean-Marc Montanier. "Environment-driven open-ended evolution with a population of autonomous robots." In: *Evolving Physical Systems Workshop*. 2012 (cited on pages 3, 8, 13, 16).

[43] Edward Hughes et al. *Open-Endedness is Essential for Artificial Superhuman Intelligence*. 2024. URL: https://arxiv.org/abs/2406.04268 (cited on pages 4, 156, 157).

[44] Open Ended Learning Team et al. "Open-ended learning leads to generally capable agents." In: *arXiv preprint arXiv:2107.12808* (2021) (cited on pages 4, 11, 12, 31, 38, 105, 107, 115, 146, 149, 151, 156).

[45] Michael Matthews et al. *Kinetix: Investigating the Training of General Agents through Open-Ended Physics-Based Control Tasks*. 2024. URL: https://arxiv.org/abs/2410.23208 (cited on pages 4, 12, 149, 156).

[46] Maxence Faldor et al. "OMNI-EPIC: Open-endedness via Models of human Notions of Interestingness with Environments Programmed in Code." In: *arXiv preprint arXiv:2405.15568* (2024) (cited on pages 4, 148, 156).

[47] Trieu H Trinh et al. "Solving olympiad geometry without human demonstrations." In: *Nature* 625.7995 (2024), pp. 476–482 (cited on pages 4, 156).

[48] Laetitia Teodorescu et al. "Codeplay: Autotelic Learning through Collaborative Self-Play in Programming Environments." In: *IMOL 2023-Intrinsically Motivated Open-ended Learning workshop at NeurIPS 2023*. 2023 (cited on pages 4, 16, 156).

[49] Jürgen Schmidhuber. "Powerplay: Training an increasingly general problem solver by continually searching for the simplest still unsolvable problem." In: *Frontiers in psychology* 4 (2013), p. 313 (cited on pages 4, 16, 156).

[50] Chris Lu et al. "Jaxlife: An open-ended agentic simulator." In: *ALIFE 2024: Proceedings of the 2024 Artificial Life Conference*. MIT Press. 2024 (cited on pages 4, 13, 147, 150).

[51] David Ha and Yujin Tang. "Collective intelligence for deep learning: A survey of recent developments." In: *Collective Intelligence* 1.1 (2022), p. 26339137221114874. DOI: 10.1177/26339137221114874 (cited on pages 4, 7, 72).

[52] Sebastian Risi. *The Future of Artificial Intelligence Is Self-Organizing and Self-Assembling*. 2021. URL: https://sebastianrisi.com/self_assembling_ai/ (visited on 05/02/2024) (cited on pages 4, 7, 72).

[53] Alexandre Variengien et al. "Towards Self-organized Control: Using Neural Cellular Automata to Robustly Control a Cart-pole Agent." In: *Innovations in Machine Intelligence* (Dec. 2021). DOI: 10.54854/imi2021.01 (cited on pages 4, 7, 36).

[54] Elias Najarro and Sebastian Risi. "Meta-learning through hebbian plasticity in random networks." In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 20719–20731 (cited on pages 4, 7, 11).

[55] Yujin Tang and David Ha. "The sensory neuron as a transformer: Permutation-invariant neural networks for reinforcement learning." In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 22574–22587 (cited on page 4).

[56] Eleni Nisioti et al. “Growing Artificial Neural Networks for Control: the Role of Neuronal Diversity.” In: *arXiv preprint arXiv:2405.08510* (2024) (cited on pages 4, 152).

[57] Erwan Plantec et al. “Evolving Self-Assembling Neural Networks: From Spontaneous Activity to Experience-Dependent Learning.” In: *ALIFE 2024: Proceedings of the 2024 Artificial Life Conference*. MIT Press. 2024 (cited on pages 4, 152).

[58] Jean-Baptiste Mouret and Jeff Clune. *Illuminating search spaces by mapping elites*. 2015. URL: https://arxiv.org/abs/1504.04909 (cited on page 4).

[59] Joel Lehman and Kenneth O. Stanley. “Evolving a diversity of virtual creatures through novelty search and local competition.” In: *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*. GECCO ’11. Dublin, Ireland: Association for Computing Machinery, 2011, pp. 211–218. DOI: 10.1145/2001576.2001606 (cited on page 4).

[60] Herbie Bradley et al. “Quality-Diversity through AI Feedback.” In: *The Twelfth International Conference on Learning Representations*. 2024 (cited on page 4).

[61] Justin K Pugh, Lisa B Soros, and Kenneth O Stanley. “Quality diversity: A new frontier for evolutionary computation.” In: *Frontiers in Robotics and AI* 3 (2016), p. 202845 (cited on page 4).

[62] Antoine Cully. “Autonomous skill discovery with quality-diversity and unsupervised descriptors.” In: *Proceedings of the Genetic and Evolutionary Computation Conference*. 2019, pp. 81–89 (cited on page 4).

[63] Joel Lehman and Kenneth O. Stanley. “Abandoning objectives: Evolution through the search for novelty alone.” In: *Evol. Comput.* 19.2 (June 2011), pp. 189–223. DOI: 10.1162/EVCO_a_00025 (cited on page 4).

[64] Julien Pourcel et al. “ACES: Generating a Diversity of Challenging Programming Puzzles with Autotelic Generative Models.” In: *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 2024 (cited on page 4).

[65] Yufei Wang et al. “Robogen: Towards unleashing infinite data for automated robot learning via generative simulation.” In: *arXiv preprint arXiv:2311.01455* (2023) (cited on pages 4, 16, 148, 156).

[66] Peter Godfrey-Smith. *Complexity and the Function of Mind in Nature*. Cambridge University Press, 1998 (cited on pages 5, 11).

[67] Humberto R Maturana and Francisco J Varela. *Autopoiesis and cognition: The realization of the living*. 1980 (cited on pages 6, 27, 30, 38).

[68] Andrew Adamatzky. “Game of Life Cellular Automata.” In: (Jan. 2010) (cited on pages 6, 27).

[69] Christopher G Langton. “Studying artificial life with cellular automata.” In: *Physica D: nonlinear phenomena* 22.1-3 (1986), pp. 120–149 (cited on page 6).

[70] Christopher G. Langton. “Self-Reproduction in Cellular Automata.” In: *Physica D: Nonlinear Phenomena* 10.1 (Jan. 1984), pp. 135–144. DOI: 10.1016/0167-2789(84)90256-2. (Visited on 02/24/2024) (cited on pages 6, 52).

[71] Hiroki Sayama. “Self-replicating worms that increase structural complexity through gene transmission.” In: (2000) (cited on pages 6–8, 15).

[72] Germán Kruszewski and Tomáš Mikolov. “Emergence of self-reproducing metabolisms as recursive algorithms in an artificial chemistry.” In: *Artificial Life* 27.3–4 (2021), pp. 277–299 (cited on pages 6–8, 15, 52).

[73] Peter Dittrich, Jens Ziegler, and Wolfgang Banzhaf. “Artificial chemistries—a review.” In: *Artificial life* 7.3 (2001), pp. 225–275 (cited on page 6).

[74] Simon J Hickinbotham et al. “Diversity from a Monoculture-Effects of Mutation-on-Copy in a String-Based Artificial Chemistry.” In: *ALife*. 2010, pp. 24–31 (cited on page 6).

[75] Deepak Pathak et al. “Learning to Control Self- Assembling Morphologies: A Study of Generalization via Modularity.” In: *NeurIPS*. 2019 (cited on pages 6, 7, 16, 153).

[76] Kazuya Horibe, Kathryn Walker, and Sebastian Risi. "Regenerating soft robots through neural cellular automata." In: *Genetic Programming: 24th European Conference, EuroGP 2021, Held as Part of EvoStar 2021, Virtual Event, April 7–9, 2021, Proceedings 24*. Springer. 2021, pp. 36–50 (cited on page 6).

[77] Randall D Beer. "Autopoiesis and cognition in the game of life." In: *Artificial Life* 10.3 (2004), pp. 309–326 (cited on pages 6, 24, 27, 32).

[78] Randall D Beer. "Bittorio revisited: structural coupling in the Game of Life." In: *Adaptive Behavior* 28.4 (2020), pp. 197–212 (cited on pages 6, 24, 32, 38).

[79] R. Beer. "Characterizing Autopoiesis in the Game of Life." In: *Artificial Life* 21 (2015), pp. 1–19 (cited on pages 6, 24).

[80] Stephen Wolfram. "Cellular Automata as Models of Complexity." In: *Nature* 311.5985 (Oct. 1984), pp. 419–424. DOI: 10.1038/311419a0. (Visited on 08/12/2022) (cited on pages 6, 24).

[81] F. J. Varela, E. Thompson, and E. Rosch. *The embodied mind*. 1991 (cited on pages 6, 24).

[82] Evan Thompson. *Mind in life: Biology, phenomenology, and the sciences of mind*. Harvard University Press, 2010 (cited on pages 6, 24).

[83] Clément Moulin-Frier et al. "Embodied artificial intelligence through distributed adaptive control: An integrated framework." In: *2017 Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*. 2017, pp. 324–330. DOI: 10.1109/DEVLRN.2017.8329825 (cited on page 7).

[84] Kazuya Horibe and Naoto Yoshida. *Emergence of Implicit World Models from Mortal Agents*. 2024. URL: https://arxiv.org/abs/2411.12304 (cited on page 7).

[85] Eleonora Bilotta, Antonio Lafusa, Pietro Pantano, et al. "Is self-replication an embedded characteristic of artificial/living matter." In: *Artificial life VIII*. Citeseer. 2002, pp. 38–48 (cited on pages 7, 8, 15).

[86] Hui-Hsien Chou and James A Reggia. "Emergence of self-replicating structures in a cellular automata space." In: *Physica D: Nonlinear Phenomena* 110.3-4 (1997), pp. 252–276 (cited on pages 7, 8, 15).

[87] Jyrki Alakuijala et al. "Computational life: How well-formed, self-replicating programs emerge from simple interaction." In: *arXiv preprint arXiv:2406.19108* (2024) (cited on pages 7, 8, 15).

[88] Lana Sinapayen. "Self-replication, spontaneous mutations, and exponential genetic drift in neural cellular automata." In: *ALIFE 2023: Ghost in the Machine: Proceedings of the 2023 Artificial Life Conference*. MIT Press. 2023 (cited on pages 7, 8, 52).

[89] PETER R. GRANT. *Ecology and Evolution of Darwin's Finches (Princeton Science Library Edition)*. REV - Revised. Princeton University Press, 1986. (Visited on 01/12/2025) (cited on page 7).

[90] Rosemary G. Gillespie, Francis G. Howarth, and George K. Roderick. "Adaptive Radiation." In: *Encyclopedia of Biodiversity*. Ed. by Simon Asher Levin. New York: Elsevier, 2001, pp. 25–44. DOI: https://doi.org/10.1016/B0-12-226865-2/00003-1 (cited on page 7).

[91] Charles Ofria and Claus O Wilke. "Avida: Evolution experiments with self-replicating computer programs." In: *Artificial Life Models in Software* (2005), pp. 3–35 (cited on pages 8, 13, 16).

[92] Sourabh Katoch, Sumit Singh Chauhan, and Vijay Kumar. "A review on genetic algorithm: past, present, and future." In: *Multimedia tools and applications* 80 (2021), pp. 8091–8126 (cited on page 8).

[93] Pedro G Espejo, Sebastián Ventura, and Francisco Herrera. "A survey on the application of genetic programming to classification." In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 40.2 (2009), pp. 121–144 (cited on page 8).

[94] William B Langdon. "Genetic programming and data structures: genetic programming+ data structures= automatic programming!" In: (1998) (cited on page 8).

[95] N. Hansen and A. Ostermeier. "Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation." In: *Proceedings of IEEE International Conference on Evolutionary Computation*. 1996, pp. 312–317. DOI: 10.1109/ICEC.1996.542381 (cited on page 8).

[96] Nikolaus Hansen and Andreas Ostermeier. "Completely derandomized self-adaptation in evolution strategies." In: *Evolutionary computation* 9.2 (2001), pp. 159–195 (cited on page 8).

[97] Stefan Kern et al. "Learning probability distributions in continuous evolutionary algorithms–a comparative review." In: *Natural Computing* 3 (2004), pp. 77–112 (cited on page 8).

[98] D.G. Mayer et al. "Survival of the fittest—genetic algorithms versus evolution strategies in the optimization of systems models." In: *Agricultural Systems* 60.2 (1999), pp. 113–122. DOI: https://doi.org/10.1016/S0308-521X(99)00022-0 (cited on page 8).

[99] Tim Salimans et al. *Evolution Strategies as a Scalable Alternative to Reinforcement Learning*. 2017. URL: https://arxiv.org/abs/1703.03864 (cited on pages 8, 53, 58, 65, 80, 83, 190).

[100] Yuqiao Liu et al. "A Survey on Evolutionary Neural Architecture Search." In: *IEEE Transactions on Neural Networks and Learning Systems* 34.2 (Feb. 2023), pp. 550–570. DOI: 10.1109/TNNLS.2021.3100554 (cited on pages 8, 80).

[101] Kenneth O. Stanley and Risto Miikkulainen. "Evolving Neural Networks through Augmenting Topologies." In: *Evolutionary Computation* 10.2 (2002), pp. 99–127. DOI: 10.1162/106365602320169811 (cited on pages 8, 152).

[102] Felipe Petroski Such et al. *Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning*. Apr. 2018. DOI: 10.48550/arXiv.1712.06567 (cited on pages 8, 80).

[103] Yujin Tang, Yingtao Tian, and David Ha. "EvoJAX: hardware-accelerated neuroevolution." In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. GECCO '22. Boston, Massachusetts: Association for Computing Machinery, 2022, pp. 308–311. DOI: 10.1145/3520304.3528770 (cited on pages 8, 151).

[104] Gianluca Baldassarre. "What are intrinsic motivations? A biological perspective." In: *2011 IEEE international conference on development and learning (ICDL)*. Vol. 2. IEEE. 2011, pp. 1–8 (cited on page 8).

[105] Gianluca Baldassarre et al. "Intrinsic motivations and open-ended development in animals, humans, and robots: an overview." In: *Frontiers in psychology* 5 (2014), p. 985 (cited on page 8).

[106] Pierre-Yves Oudeyer and Linda B Smith. "How evolution may work through curiosity-driven developmental process." In: *Topics in Cognitive Science* 8.2 (2016), pp. 492–502 (cited on page 8).

[107] Pierre-Yves Oudeyer, Frdric Kaplan, and Verena V Hafner. "Intrinsic motivation systems for autonomous mental development." In: *IEEE transactions on evolutionary computation* 11.2 (2007), pp. 265–286 (cited on page 8).

[108] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018 (cited on pages 8, 92, 121).

[109] David Silver et al. "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play." In: *Science* 362.6419 (2018), pp. 1140–1144. DOI: 10.1126/science.aar6404 (cited on pages 9, 16).

[110] David Silver et al. "Mastering the game of Go without human knowledge." In: *Nature* 550.7676 (Oct. 2017). Publisher: Nature Publishing Group, pp. 354–359. DOI: 10.1038/nature24270. (Visited on 12/17/2018) (cited on page 9).

[111] Volodymyr Mnih. "Playing atari with deep reinforcement learning." In: *arXiv preprint arXiv:1312.5602* (2013) (cited on page 9).

[112] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning." In: *nature* 518.7540 (2015), pp. 529–533 (cited on pages 9, 110).

[113] Oriol Vinyals et al. “Grandmaster level in StarCraft II using multi-agent reinforcement learning.” In: *Nature* 575.7782 (2019), pp. 350–354 (cited on pages 9, 16, 106).

[114] Christopher Berner et al. “Dota 2 with large scale deep reinforcement learning.” In: *arXiv preprint arXiv:1912.06680* (2019) (cited on pages 9, 16, 105, 106).

[115] Max Jaderberg et al. “Human-level performance in 3D multiplayer games with population-based reinforcement learning.” In: *Science* 364.6443 (2019), pp. 859–865 (cited on pages 9, 16, 105, 106, 118).

[116] OpenAI: Marcin Andrychowicz et al. “Learning dexterous in-hand manipulation.” In: *The International Journal of Robotics Research* 39.1 (2020), pp. 3–20 (cited on page 9).

[117] Ilge Akkaya et al. “Solving rubik’s cube with a robot hand.” In: *arXiv preprint arXiv:1910.07113* (2019) (cited on pages 9, 31, 107).

[118] Jemin Hwangbo et al. “Learning agile and dynamic motor skills for legged robots.” In: *Science Robotics* 4.26 (2019), eaau5872 (cited on page 9).

[119] Xue Bin Peng et al. “Sim-to-real transfer of robotic control with dynamics randomization.” In: *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2018, pp. 3803–3810 (cited on page 9).

[120] Yecheng Jason Ma et al. “Eureka: Human-level reward design via coding large language models.” In: *arXiv preprint arXiv:2310.12931* (2023) (cited on page 9).

[121] Yecheng Jason Ma et al. “DrEureka: Language Model Guided Sim-To-Real Transfer.” In: *arXiv preprint arXiv:2406.01967* (2024) (cited on page 9).

[122] Matthias Plappert et al. “Multi-goal reinforcement learning: Challenging robotics environments and request for research.” In: *arXiv preprint arXiv:1802.09464* (2018) (cited on page 9).

[123] Minghuan Liu, Menghui Zhu, and Weinan Zhang. “Goal-conditioned reinforcement learning: Problems and solutions.” In: *arXiv preprint arXiv:2201.08299* (2022) (cited on pages 9, 118).

[124] Shibhansh Dohare et al. “Loss of plasticity in deep continual learning.” In: *Nature* 632.8026 (2024), pp. 768–774 (cited on page 9).

[125] Zaheer Abbas et al. “Loss of plasticity in continual deep reinforcement learning.” In: *Conference on Lifelong Learning Agents*. PMLR. 2023, pp. 620–636 (cited on page 9).

[126] Arthur Juliani and Jordan T. Ash. “A Study of Plasticity Loss in On-Policy Deep Reinforcement Learning.” In: *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 2024 (cited on page 9).

[127] Clare Lyle et al. “Understanding plasticity in neural networks.” In: *International Conference on Machine Learning*. PMLR. 2023, pp. 23190–23211 (cited on page 9).

[128] Timo Klein et al. “Plasticity Loss in Deep Reinforcement Learning: A Survey.” In: *arXiv preprint arXiv:2411.04832* (2024) (cited on page 9).

[129] Clare Lyle et al. “Disentangling the causes of plasticity loss in neural networks.” In: *arXiv preprint arXiv:2402.18762* (2024) (cited on page 9).

[130] Evgenii Nikishin et al. “Deep reinforcement learning with plasticity injection.” In: *Advances in Neural Information Processing Systems* 36 (2024) (cited on page 9).

[131] James Kirkpatrick et al. “Overcoming catastrophic forgetting in neural networks.” In: *Proceedings of the national academy of sciences* 114.13 (2017), pp. 3521–3526 (cited on page 9).

[132] Robert M. French. “Catastrophic forgetting in connectionist networks.” In: *Trends in Cognitive Sciences* 3.4 (1999), pp. 128–135. DOI: https://doi.org/10.1016/S1364-6613(99)01294-2 (cited on page 9).

[133] Pawel Ladosz et al. “Exploration in deep reinforcement learning: A survey.” In: *Information Fusion* 85 (2022), pp. 1–22 (cited on page 9).

[134] Alexandr Ten, Pierre-Yves Oudeyer, and Clément Moulin-Frier. "Curiosity-driven exploration: Diversity of mechanisms and functions." In: *The Drive for Knowledge: The Science of Human Information Seeking*. 2022. DOI: 10.1017/9781009026949 (cited on pages 9, 101, 151).

[135] Pierre-Yves Oudeyer and Frederic Kaplan. "What is intrinsic motivation? A typology of computational approaches." In: *Frontiers in neurorobotics* 1 (2007), p. 108 (cited on pages 9, 118, 119).

[136] Marc Bellemare et al. "Unifying count-based exploration and intrinsic motivation." In: *Advances in neural information processing systems* 29 (2016) (cited on pages 9, 101).

[137] Haoran Tang et al. "# exploration: A study of count-based exploration for deep reinforcement learning." In: *Advances in neural information processing systems* 30 (2017) (cited on pages 9, 101).

[138] Yuri Burda et al. "Exploration by random network distillation." In: *arXiv preprint arXiv:1810.12894* (2018) (cited on pages 9, 101).

[139] Yuri Burda et al. "Large-scale study of curiosity-driven learning." In: *arXiv preprint arXiv:1808.04355* (2018) (cited on pages 9, 101).

[140] Deepak Pathak et al. "Curiosity-driven exploration by self-supervised prediction." In: *International conference on machine learning*. PMLR. 2017, pp. 2778–2787 (cited on pages 9, 101, 120).

[141] Glen Berseth et al. "Smirl: Surprise minimizing reinforcement learning in unstable environments." In: *arXiv preprint arXiv:1912.05510* (2019) (cited on pages 9, 101).

[142] Nicolas Bougie and Ryutaro Ichise. "Skill-based curiosity for intrinsically motivated reinforcement learning." In: *Machine Learning* 109 (2020), pp. 493–512 (cited on pages 9, 101).

[143] Christoph Salge, Cornelius Glackin, and Daniel Polani. "Changing the environment based on empowerment as intrinsic motivation." In: *Entropy* 16.5 (2014), pp. 2789–2819 (cited on pages 9, 101).

[144] Cédric Colas et al. "Autotelic agents with intrinsically motivated goal-conditioned reinforcement learning: a short survey." In: *Journal of Artificial Intelligence Research* 74 (2022), pp. 1159–1199 (cited on pages 9, 101, 118–120, 122, 132).

[145] Luc Steels. "The autotelic principle." In: *Embodied artificial intelligence*. Springer, 2004, pp. 231–242 (cited on pages 9, 101).

[146] Claire Kramsch. "Language and culture." In: *AILA review* 27.1 (2014), pp. 30–55 (cited on page 9).

[147] Jonathan Paige and Charles Perreault. "3.3 million years of stone tool complexity suggests that cumulative culture began during the Middle Pleistocene." In: *Proceedings of the National Academy of Sciences* 121.26 (2024), e2319175121 (cited on page 10).

[148] Alex Mesoudi and Andrew Whiten. "The multiple roles of cultural transmission experiments in understanding human cultural evolution." In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 363.1509 (2008), pp. 3489–3501 (cited on page 10).

[149] Kenny Smith, Henry Brighton, and Simon Kirby. "Complex systems in language evolution: the cultural emergence of compositional structure." In: *Advances in complex systems* 6.04 (2003), pp. 537–558 (cited on page 10).

[150] Jonathan Cook et al. "Artificial Generational Intelligence: Cultural Accumulation in Reinforcement Learning." In: *arXiv preprint arXiv:2406.00392* (2024) (cited on pages 10, 100, 154).

[151] Cultural General Intelligence Team et al. "Learning Robust Real-Time Cultural Transmission without Human Data." In: *arXiv preprint arXiv:2203.00715* (2022) (cited on pages 10, 100).

[152] Kamal K Ndousse et al. "Emergent social learning via multi-agent reinforcement learning." In: *International Conference on Machine Learning*. PMLR. 2021, pp. 7991–8004 (cited on pages 10, 100, 132).

[153] Grgur Kovač et al. "The SocialAI school: a framework leveraging developmental psychology toward artificial socio-cultural agents." In: *Frontiers in Neurorobotics* 18 (2024), p. 1396359 (cited on page 10).

[154] Eleni Nisioti et al. "Social network structure shapes innovation: experience-sharing in RL with SAPIENS." In: *arXiv preprint arXiv:2206.05060* (2022) (cited on pages 10, 102).

[155] Yoshua Bengio. "Evolving Culture Versus Local Minima." In: *Growing Adaptive Machines: Combining Development and Learning in Artificial Neural Networks*. Ed. by Taras Kowaliw, Nicolas Bredeche, and René Doursat. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 109–138. DOI: 10.1007/978-3-642-55337-0_3 (cited on page 10).

[156] Levin Brinkmann et al. "Machine culture." In: *Nature Human Behaviour* 7.11 (2023), pp. 1855–1868 (cited on page 10).

[157] Alberto Acerbi and Joseph M Stubbersfield. "Large language models show human-like content biases in transmission chain experiments." In: *Proceedings of the National Academy of Sciences* 120.44 (2023), e2313790120 (cited on page 10).

[158] Jérémy Perez et al. *When LLMs Play the Telephone Game: Cumulative Changes and Attractors in Iterated Cultural Transmissions*. 2024. URL: https://arxiv.org/abs/2407.04503 (cited on page 10).

[159] James M Borg et al. "Evolved open-endedness in cultural evolution: A new dimension in open-ended evolution research." In: *Artificial Life* 30.3 (2024), pp. 417–438 (cited on page 10).

[160] David W Stephens. "Change, regularity, and value in the evolution of animal learning." In: *Behavioral Ecology* 2.1 (1991), pp. 77–89 (cited on pages 10, 101).

[161] Timothy D Johnston. "Selective costs and benefits in the evolution of learning." In: *Advances in the Study of Behavior*. Vol. 12. Elsevier, 1982, pp. 65–106 (cited on page 10).

[162] Albano Beja-Pereira et al. "Gene-culture coevolution between cattle milk protein genes and human lactase genes." In: *Nature genetics* 35.4 (2003), pp. 311–313 (cited on page 10).

[163] Olivier Sigaud. "Combining evolution and deep reinforcement learning for policy search: A survey." In: *ACM Transactions on Evolutionary Learning* 3.3 (2023), pp. 1–20 (cited on page 10).

[164] Massimiliano Schembri, Marco Mirolli, and Gianluca Baldassarre. "Evolution and learning in an intrinsically motivated reinforcement learning robot." In: *Advances in Artificial Life: 9th European Conference, ECAL 2007, Lisbon, Portugal, September 10-14, 2007. Proceedings 9*. Springer. 2007, pp. 294–303 (cited on page 10).

[165] Max Jaderberg et al. *Population Based Training of Neural Networks*. 2017. URL: https://arxiv.org/abs/1711.09846 (cited on page 11).

[166] Aloïs Pourchot and Olivier Sigaud. "CEM-RL: Combining evolutionary and gradient-based methods for policy search." In: *arXiv preprint arXiv:1810.01222* (2018) (cited on page 11).

[167] Olle Nilsson and Antoine Cully. "Policy gradient assisted map-elites." In: *Proceedings of the Genetic and Evolutionary Computation Conference*. 2021, pp. 866–875 (cited on pages 11, 25).

[168] Thomas Pierrot et al. "Diversity policy gradient for sample efficient quality-diversity optimization." In: *Proceedings of the Genetic and Evolutionary Computation Conference*. 2022, pp. 1075–1083 (cited on pages 11, 25).

[169] Anna Vettoruzzo et al. "Advances and challenges in meta-learning: A technical review." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024) (cited on pages 11, 102).

[170] Jacob Beck et al. *A Survey of Meta-Reinforcement Learning*. 2024. URL: https://arxiv.org/abs/2301.08028 (cited on pages 11, 102).

[171] Ferran Alet* et al. "Meta-learning curiosity algorithms." In: *International Conference on Learning Representations*. 2020 (cited on pages 11, 102).

[172] Sebastian Thrun and Lorien Pratt. "Learning to learn: Introduction and overview." In: *Learning to learn*. Springer, 1998, pp. 3–17 (cited on pages 11, 102).

[173] Timothy Hospedales et al. "Meta-learning in neural networks: A survey." In: *IEEE transactions on pattern analysis and machine intelligence* 44.9 (2021), pp. 5149–5169 (cited on pages 11, 102).

[174] Chris Lu et al. "Discovered policy optimisation." In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 16455–16468 (cited on pages 11, 151, 156).

[175] Louis Kirsch, Sjoerd van Steenkiste, and Juergen Schmidhuber. "Improving Generalization in Meta Reinforcement Learning using Learned Objectives." In: *International Conference on Learning Representations*. 2020 (cited on page 11).

[176] Junhyuk Oh et al. "Discovering reinforcement learning algorithms." In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 1060–1070 (cited on page 11).

[177] Louis Kirsch et al. "General-purpose in-context learning by meta-learning transformers." In: *arXiv preprint arXiv:2212.04458* (2022) (cited on page 11).

[178] Jiaoda Li et al. "What Do Language Models Learn in Context? The Structured Task Hypothesis." In: *arXiv preprint arXiv:2406.04216* (2024) (cited on page 11).

[179] Nicolas Schweighofer and Kenji Doya. "Meta-learning in Reinforcement Learning." In: *Neural Networks* 16.1 (2003), pp. 5–9. DOI: https://doi.org/10.1016/S0893-6080(02)00228-9 (cited on page 11).

[180] Yuji Kanagawa and Kenji Doya. "Evolution of Rewards for Food and Motor Action by Simulating Birth and Death." In: *ALIFE 2024: Proceedings of the 2024 Artificial Life Conference*. MIT Press. 2024 (cited on page 11).

[181] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks." In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 1126–1135 (cited on page 11).

[182] Jane X Wang et al. *Learning to reinforcement learn*. 2017 (cited on pages 11, 13, 100, 102, 106, 137, 138).

[183] Yan Duan et al. "Rl ^2: Fast reinforcement learning via slow reinforcement learning." In: *arXiv preprint arXiv:1611.02779* (2016) (cited on pages 11, 13, 100, 102, 106, 132, 137, 138).

[184] Adaptive Agent Team et al. "Human-timescale adaptation in an open-ended task space." In: *arXiv preprint arXiv:2301.07608* (2023) (cited on pages 11, 13, 83, 100, 102, 105, 107, 109, 115, 136–139, 146, 147, 149, 151).

[185] Louis Kirsch et al. "Towards general-purpose in-context learning agents." In: Workshop on Distribution Shifts, 37th Conference on Neural Information ... 2023 (cited on pages 11, 100, 102).

[186] Robert Tjarko Lange and Henning Sprekeler. "Learning not to learn: Nature versus nurture in silico." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 7. 2022, pp. 7290–7299 (cited on pages 11, 13, 100, 102).

[187] Anil K Seth. "Agent-based modelling and the environmental complexity thesis." In: (2002) (cited on page 11).

[188] Margaret Evans et al. "Insights on the Evolution of Plant Succulence from a Remarkable Radiation in Madagascar (Euphorbia)." In: *Systematic Biology* 63.5 (May 2014), pp. 697–711. DOI: 10.1093/sysbio/syu035 (cited on page 12).

[189] Howard Griffiths and Jamie Males. "Succulent plants." In: *Current Biology* 27.17 (2017), R890–R896 (cited on page 12).

[190] Mark A Maslin, Susanne Shultz, and Martin H Trauth. "A synthesis of the theories and concepts of early human evolution." In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 370.1663 (2015), p. 20140064 (cited on pages 12, 13).

[191] Gillian R Brown et al. *Evolutionary accounts of human behavioural diversity*. 2011 (cited on page 12).

[192] Rebecca Sear, David W Lawson, and Thomas E Dickins. "Synthesis in the human evolutionary behavioural sciences." In: *Journal of Evolutionary Psychology* 5.1 (2007), pp. 3–28 (cited on page 12).

[193] Richard Potts. "Hominin evolution in settings of strong environmental variability." In: *Quaternary Science Reviews* 73 (2013), pp. 1–13 (cited on pages 12, 13).

[194] Evan Zheran Liu et al. "Simple embodied language learning as a byproduct of meta-reinforcement learning." In: *International Conference on Machine Learning*. PMLR. 2023, pp. 21997–22008 (cited on page 12).

[195] Michael Dennis et al. “Emergent complexity and zero-shot transfer via unsupervised environment design.” In: *Advances in neural information processing systems* 33 (2020), pp. 13049–13061 (cited on pages 12, 13, 15, 16, 150).

[196] Niels Justesen et al. “Illuminating generalization in deep reinforcement learning through procedural level generation.” In: *arXiv preprint arXiv:1806.10729* (2018) (cited on pages 12, 107, 150).

[197] Rui Wang et al. “Paired Open-Ended Trailblazer (POET): Endlessly Generating Increasingly Complex and Diverse Learning Environments and Their Solutions.” In: *arXiv:1901.01753 [cs]* (2019). arXiv: 1901.01753. (Visited on 04/21/2020) (cited on pages 13, 15, 31).

[198] Rémy Portelas et al. “Automatic curriculum learning for deep RL: a short survey.” In: *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*. 2021, pp. 4819–4825 (cited on pages 13, 15, 31).

[199] Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. “Prioritized level replay.” In: *International Conference on Machine Learning*. PMLR. 2021, pp. 4940–4950 (cited on pages 13, 15, 150).

[200] Jean-Marc Montanier and Nicolas Bredeche. “Surviving the tragedy of commons: emergence of altruism in a population of evolving autonomous agents.” In: *European conference on artificial life*. 2011 (cited on page 13).

[201] Ettore Randazzo and Alexander Mordvintsev. “Biomaker CA: a Biome Maker project using Cellular Automata.” In: *arXiv preprint arXiv:2307.09320* (2023) (cited on pages 13, 150).

[202] Richard C Lewontin. “The organism as the subject and object of evolution.” In: *Scientia* 77.18 (1983) (cited on pages 13, 74).

[203] F John Odling-Smee, Kevin N Laland, and Marcus W Feldman. “Niche construction.” In: *The American Naturalist* 147.4 (1996), pp. 641–648 (cited on pages 13, 14, 74).

[204] Kevin N. Laland et al. “The extended evolutionary synthesis: its structure, assumptions and predictions.” In: *Proceedings of the Royal Society B: Biological Sciences* 282.1813 (Aug. 2015). Publisher: Royal Society, p. 20151019. DOI: 10.1098/rspb.2015.1019. (Visited on 09/05/2022) (cited on pages 13, 14, 74).

[205] Axel Constant et al. “A variational approach to niche construction.” In: *Journal of the Royal Society Interface* 15.141 (2018), p. 20170685 (cited on pages 13, 74).

[206] David C Krakauer, Karen M Page, and Douglas H Erwin. “Diversity, dilemmas, and monopolies of niche construction.” In: *The American Naturalist* 173.1 (2009), pp. 26–40 (cited on pages 13, 74).

[207] Kevin Laland, John Odling-Smee, and John Endler. “Niche construction, sources of selection and trait coevolution.” In: *Interface Focus* 7.5 (2017), p. 20160147 (cited on pages 13, 74).

[208] Clive G. Jones, John H. Lawton, and Moshe Shachak. “Organisms as Ecosystem Engineers.” In: *Oikos* 69.3 (1994), pp. 373–386. (Visited on 01/08/2025) (cited on pages 13, 74).

[209] John Odling-Smee. “Niche inheritance: a possible basis for classifying multiple inheritance systems in evolution.” In: *Biological Theory* 2 (2007), pp. 276–289 (cited on pages 14, 74).

[210] Victor J Thannickal. *Oxygen in the evolution of complex life and the price we pay*. 2009 (cited on page 14).

[211] Douglas H Erwin. “Macroevolution of ecosystem engineering, niche construction and diversity.” In: *Trends in ecology & evolution* 23.6 (2008), pp. 304–310 (cited on page 14).

[212] John D Co-Reyes et al. “Ecological reinforcement learning.” In: *arXiv preprint arXiv:2006.12478* (2020) (cited on pages 14, 80).

[213] Aaron Dharna and Julian Togelius. “Co-generation of game levels and game-playing agents.” In: *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*. Vol. 16. 1. 2020, pp. 203–209 (cited on page 15).

[214] Joel Z Leibo et al. “Autocurricula and the emergence of innovation from social interaction: A manifesto for multi-agent intelligence research.” In: *arXiv preprint arXiv:1903.00742* (2019) (cited on page 15).

[215] Richard Dawkins and John Richard Krebs. “Arms races between and within species.” In: *Proceedings of the Royal Society of London. Series B. Biological Sciences* 205.1161 (1979), pp. 489–511 (cited on pages 15, 75).

[216] Luis Zaman et al. “Coevolution drives the emergence of complex traits and promotes evolvability.” In: *PLoS Biology* 12.12 (2014), e1002023 (cited on pages 15, 75, 155).

[217] Leigh Van Valen. “A New Evolutionary Law.” In: *Evolutionary Theory* 1 (1973), pp. 1–30 (cited on page 15).

[218] John L Maron, Anurag A Agrawal, and Douglas W Schemske. “Plant–herbivore coevolution and plant speciation.” In: *Ecology* 100.7 (2019), e02704 (cited on page 15).

[219] John A Gatehouse. “Plant resistance towards insect herbivores: a dynamic interaction.” In: *New phytologist* 156.2 (2002), pp. 145–169 (cited on page 15).

[220] Paul R Ehrlich and Peter H Raven. “Butterflies and plants: a study in coevolution.” In: *Evolution* (1964), pp. 586–608 (cited on page 15).

[221] Curtis M Lively, Clark Craddock, and Robert C Vrijenhoek. “Red Queen hypothesis supported by parasitism in sexual and clonal fish.” In: *Nature* 344.6269 (1990), pp. 864–866 (cited on page 15).

[222] Arthur L Samuel. “Some studies in machine learning using the game of checkers.” In: *IBM Journal of research and development* 3.3 (1959), pp. 210–229 (cited on page 16).

[223] Gerald Tesauro et al. “Temporal difference learning and TD-Gammon.” In: *Communications of the ACM* 38.3 (1995), pp. 58–68 (cited on page 16).

[224] David Silver et al. “Mastering the game of Go with deep neural networks and tree search.” In: *nature* 529.7587 (2016), pp. 484–489 (cited on pages 16, 106).

[225] Ian Goodfellow et al. “Generative adversarial nets.” In: *Advances in neural information processing systems* 27 (2014) (cited on page 16).

[226] Stefano Nolfi and Dario Floreano. “Coevolving predator and prey robots: Do “arms races” arise in artificial evolution?” In: *Artificial life* 4.4 (1998), pp. 311–335 (cited on page 16).

[227] Craig W Reynolds. “Flocks, herds and schools: A distributed behavioral model.” In: *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. 1987, pp. 25–34 (cited on page 16).

[228] Akira Shimizu et al. “Fine-tuned bee-flower coevolutionary state hidden within multiple pollination interactions.” In: *Scientific Reports* 4.1 (2014), p. 3988 (cited on page 16).

[229] Steven D Johnson and Bruce Anderson. “Coevolution between food-rewarding flowers and their pollinators.” In: *Evolution: Education and Outreach* 3 (2010), pp. 32–39 (cited on page 16).

[230] Eleni Nisioti, Katia Jodogne-del Litto, and Clément Moulin-Frier. “Grounding an Ecological Theory of Artificial Intelligence in Human Evolution.” In: *NeurIPS 2021 - Conference on Neural Information Processing Systems / Workshop: Ecological Theory of Reinforcement Learning*. virtual event, France, Dec. 2021. (Visited on 08/19/2022) (cited on pages 16, 70).

[231] Emilio Parisotto et al. “Stabilizing transformers for reinforcement learning.” In: *Proceedings of the 37th International Conference on Machine Learning*. ICML’20. JMLR.org, 2020 (cited on pages 21, 135).

[232] Michael Matthews et al. “Craftax: a lightning-fast benchmark for open-ended reinforcement learning.” In: *Proceedings of the 41st International Conference on Machine Learning*. ICML’24. Vienna, Austria: JMLR.org, 2025 (cited on pages 21, 135, 147, 215, 216).

[233] Stephen Wolfram. “Universality and complexity in cellular automata.” In: *Physica D: Nonlinear Phenomena* 10.1 (1984), pp. 1–35. DOI: *https://doi.org/10.1016/0167-2789(84)90245-8* (cited on pages 24, 27).

[234] Chris Reinke, Mayalen Etcheverry, and Pierre-Yves Oudeyer. “Intrinsically Motivated Discovery of Diverse Patterns in Self-Organizing Systems.” In: *International Conference on Learning Representations*. 2020 (cited on pages 25, 32, 37, 158, 178).

[235] Mayalen Etcheverry, Clément Moulin-Frier, and Pierre-Yves Oudeyer. "Hierarchically Organized Latent Modules for Exploratory Search in Morphogenetic Systems." In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 4846–4859 (cited on pages 25, 27, 32, 33, 158).

[236] Ricard Solé et al. "Fundamental constraints to the logic of living systems." In: *Interface Focus* 14.5 (2024), p. 20240010 (cited on page 25).

[237] David Krakauer et al. "The Information Theory of Individuality." In: *Theory in Biosciences* 139 (June 2020), pp. 209–223. DOI: *10.1007/s12064-020-00313-7* (cited on pages 30, 32, 49, 71).

[238] Ezequiel A Di Paolo. "Process and individuation: the development of sensorimotor agency." In: *Human Development* 63.3-4 (2019), pp. 202–226 (cited on pages 30–32, 41).

[239] Pamela Lyon et al. *Reframing cognition: getting down to biological basics*. 2021 (cited on pages 30, 153).

[240] Rolf Pfeifer and Josh Bongard. *How the body shapes the way we think: a new view of intelligence*. MIT press, 2006 (cited on pages 31, 46, 152).

[241] Tom Froese and Tom Ziemke. "Enactive artificial intelligence: Investigating the systemic organization of life and mind." In: *Artificial intelligence* 173.3-4 (2009), pp. 466–500 (cited on page 32).

[242] F.G. Varela, H.R. Maturana, and R. Uribe. "Autopoiesis: The organization of living systems, its characterization and a model." In: *Biosystems* 5.4 (1974), pp. 187–196. DOI: *https://doi.org/10.1016/0303-2647(74)90031-8* (cited on page 32).

[243] Barry McMullin. "Thirty years of computational autopoiesis: A review." In: *Artificial life* 10.3 (2004), pp. 277–295 (cited on page 32).

[244] Eran Agmon, Alexander J Gates, and Randall D Beer. "Ontogeny and adaptivity in a model protocell." In: *Artificial Life Conference Proceedings 13*. MIT Press. 2015, pp. 216–223 (cited on page 32).

[245] Martin Biehl, Takashi Ikegami, and Daniel Polani. "Towards Information Based Spatiotemporal Patterns as a Foundation for Agent Representation in Dynamical Systems." In: *Proceedings of the Artificial Life Conference 2016*. Cancun, Mexico: MIT Press, 2016, pp. 722–729. DOI: *10.7551/978-0-262-33936-0-ch115* (cited on pages 32, 49).

[246] Randall D Beer. "The cognitive domain of a glider in the game of life." In: *Artificial life* 20.2 (2014), pp. 183–206. DOI: *10.1162/artl_a_00125* (cited on pages 32, 33, 45).

[247] *Resilient Life: An Exploration of Perturbed Autopoietic Patterns in Conway's Game of Life*. Vol. ALIFE 2020: The 2020 Conference on Artificial Life. July 2020, pp. 656–664. DOI: *10.1162/isal_a_00305* (cited on page 32).

[248] Adrien Baranes and Pierre-Yves Oudeyer. "Active learning of inverse models with intrinsically motivated goal exploration in robots." In: *Robotics and Autonomous Systems* 61.1 (2013), pp. 49–73 (cited on page 32).

[249] Sébastien Forestier et al. "Intrinsically Motivated Goal Exploration Processes with Automatic Curriculum Learning." In: *Journal of Machine Learning Research* 23 (2022), pp. 1–41 (cited on pages 32, 35, 47, 168).

[250] Cédric Colas et al. "Curious: intrinsically motivated modular multi-goal reinforcement learning." In: *International conference on machine learning*. PMLR. 2019, pp. 1331–1340 (cited on pages 32, 120, 122, 126, 205, 207).

[251] Cédric Colas et al. "Language as a cognitive tool to imagine goals in curiosity driven exploration." In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 3761–3774 (cited on pages 32, 121, 147).

[252] Jonathan Grizou et al. "A Curious Formulation Robot Enables the Discovery of a Novel Protocell Behavior." In: *Science Advances* 6.5 (Jan. 2020), eaay4237. DOI: *10.1126/sciadv.aay4237*. (Visited on 04/23/2023) (cited on page 32).

[253] Martin J. Falk et al. *Curiosity-Driven Search for Novel Non-Equilibrium Behaviors*. Mar. 2023. (Visited on 04/28/2023) (cited on page 32).

[254] Hiroaki Kitano. “Biological robustness.” In: *Nature Reviews Genetics* 5.11 (2004), pp. 826–837 (cited on page 33).

[255] Jacqueline Gottlieb and Pierre-Yves Oudeyer. “Towards a neuroscience of active sampling and curiosity.” In: *Nature Reviews Neuroscience* 19.12 (2018), pp. 758–770 (cited on page 35).

[256] Alexander Mordvintsev et al. “Thread: Differentiable Self-organizing Systems.” In: *Distill* (2020). https://distill.pub/2020/selforg. DOI: *10.23915/distill.00027* (cited on page 36).

[257] Alexander Mordvintsev et al. “Growing Neural Cellular Automata.” In: *Distill* (2020). https://distill.pub/2020/growing ca. DOI: *10.23915/distill.00023* (cited on pages 36, 43, 52, 170).

[258] Eyvind Niklasson et al. “Self-Organising Textures.” In: *Distill* (2021). https://distill.pub/selforg/2021/textures. DOI: *10.23915/distill.00027.003* (cited on page 36).

[259] Ettore Randazzo et al. “Self-classifying MNIST Digits.” In: *Distill* (2020). https://distill.pub/2020/selforg/mnist. DOI: *10.23915/distill.00027.002* (cited on page 36).

[260] Laura N Vandenberg, Dany S Adams, and Michael Levin. “Normalized shape and location of perturbed craniofacial structures in the Xenopus tadpole reveal an innate ability to achieve correct morphology.” In: *Developmental Dynamics* 241.5 (2012), pp. 863–878 (cited on pages 41, 43).

[261] Gerhard Fankhauser. “Maintenance of normal structure in heteroploid salamander larvae, through compensation of changes in cell size by adjustment of cell number and cell shape.” In: *Journal of Experimental Zoology* 100.3 (1945), pp. 445–455 (cited on pages 41, 44).

[262] Jörg Stelling et al. “Robustness of cellular functions.” In: *Cell* 118.6 (2004), pp. 675–685 (cited on page 41).

[263] Michael Levin. “The computational boundary of a “self”: developmental bioelectricity drives multicellularity and scale-free cognition.” In: *Frontiers in Psychology* 10 (2019), p. 2688 (cited on page 45).

[264] Michael Levin. “Technological approach to mind everywhere: an experimentally-grounded framework for understanding diverse bodies and minds.” In: *Frontiers in Systems Neuroscience* (2022), p. 17 (cited on pages 45, 153).

[265] Wei Li et al. “Light-Driven and Light-Guided Microswimmers.” In: *Advanced Functional Materials* 26.18 (2016), pp. 3164–3171 (cited on page 46).

[266] Trapit Bansal et al. “Emergent Complexity via Multi-Agent Competition.” In: *CoRR* abs/1710.03748 (2017) (cited on page 49).

[267] Bert Wang-Chak Chan. “Towards Large-Scale Simulations of Open-Ended Evolution in Continuous Cellular Automata.” In: *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*. GECCO ’23 Companion. New York, NY, USA: Association for Computing Machinery, July 2023, pp. 127–130. DOI: *10.1145/3583133.3590670*. (Visited on 03/10/2024) (cited on pages 50, 59).

[268] Giovanni Pezzulo and Michael Levin. “Re-membering the body: applications of computational neuroscience to the top-down control of regeneration of limbs and other complex organs.” In: *Integrative Biology* 7.12 (2015), pp. 1487–1517 (cited on pages 50, 158).

[269] Giovanni Pezzulo and Michael Levin. “Top-down models in biology: explanation and control of complex living systems above the molecular level.” In: *Journal of The Royal Society Interface* 13.124 (2016), p. 20160555 (cited on pages 50, 158).

[270] Reiichiro Nakano et al. “Webgpt: Browser-assisted question-answering with human feedback.” In: *arXiv preprint arXiv:2112.09332* (2021) (cited on page 50).

[271] Timo Schick et al. “Toolformer: Language models can teach themselves to use tools.” In: *arXiv preprint arXiv:2302.04761* (2023) (cited on page 50).

[272] Sam Kriegman et al. “A scalable pipeline for designing reconfigurable organisms.” In: *Proceedings of the National Academy of Sciences* 117.4 (2020), pp. 1853–1859 (cited on page 50).

[273] Mo R Ebrahimkhani and Michael Levin. "Synthetic living machines: A new window on life." In: *Iscience* (2021), p. 102505 (cited on page 50).

[274] Kenneth O. Stanley. "Why Open-Endedness Matters." In: *Artificial Life* 25.3 (Aug. 2019), pp. 232–235. DOI: 10.1162/artl_a_00294. (Visited on 09/06/2022) (cited on page 52).

[275] John Von Neumann and Arthur W. (Arthur Walter) Burks. *Theory of Self-Reproducing Automata*. Urbana, University of Illinois Press, 1966. (Visited on 08/12/2022) (cited on page 52).

[276] Tim J. Hutton. "Codd's Self-Replicating Computer." In: *Artificial Life* 16.2 (2010), pp. 99–117. DOI: 10.1162/artl.2010.16.2.16200 (cited on page 52).

[277] Hiroki Sayama. "Toward the Realization of an Evolving Ecosystem on Cellular Automata." In: *Proc. Fourth Int. Symp. Artificial Life and Robotics* (1999), pp. 254–257. (Visited on 02/01/2023) (cited on page 52).

[278] Hiroki Sayama and Chrystopher L. Nehaniv. *Self-Reproduction and Evolution in Cellular Automata: 25 Years after Evoloops*. Feb. 2024. DOI: 10.48550/arXiv.2402.03961. (Visited on 02/28/2024) (cited on page 52).

[279] Mark A. Bedau and Norman H. Packard. *Measurement of Evolutionary Activity, Teleology, and Life*. 1996 (cited on page 53).

[280] Alastair Droop and Simon Hickinbotham. "A Quantitative Measure of Non-Neutral Evolutionary Activity for Systems That Exhibit Intrinsic Fitness." In: *ALIFE 2012: The Thirteenth International Conference on the Synthesis and Simulation of Living Systems*. MIT Press, July 2012, pp. 45–52. DOI: 10.1162/978-0-262-31050-5-ch007 (cited on pages 53, 61, 68, 70).

[281] T.S. Ray and J. Hart. "Evolution of Differentiated Multi-Threaded Digital Organisms." In: *Proceedings 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human and Environment Friendly Robots with High Intelligence and Emotional Quotients (Cat. No.99CH36289)*. Vol. 1. Oct. 1999, 1–10 vol.1. DOI: 10.1109/IROS.1999.812972 (cited on page 53).

[282] Chris Adami and C. Titus Brown. *Evolutionary Learning in the 2D Artificial Life System "Avida"*. May 1994. DOI: 10.48550/arXiv.adap-org/9405003 (cited on page 53).

[283] M. Bedau et al. "A Comparison of Evolutionary Activity in Artificial Evolving Systems and in the Biosphere." In: Mar. 1998. (Visited on 02/29/2024) (cited on page 53).

[284] Simon Hickinbotham and Susan Stepney. "Conservation of Matter Increases Evolutionary Activity." In: *ECAL 2015: The 13th European Conference on Artificial Life*. MIT Press, July 2015, pp. 98–105. DOI: 10.1162/978-0-262-33027-5-ch024 (cited on page 53).

[285] Gennady Shkliarevsky. "Conservation, Creation, and Evolution: Revising the Darwinian Project." In: *Journal of Evolutionary Science* 1.2 (Sept. 2019), pp. 1–30. DOI: 10.14302/issn.2689-4602.jes-19-2990. (Visited on 02/29/2024) (cited on page 53).

[286] Stuart Bartlett and Michael L. Wong. "Defining Lyfe in the Universe: From Three Privileged Functions to Four Pillars." In: *Life* 10.4 (Apr. 2020), p. 42. DOI: 10.3390/life10040042. (Visited on 02/29/2024) (cited on page 53).

[287] Mykhailo Moroz. *Reintegration Tracking*. https://michaelmoroz.github.io/Reintegration-Tracking/. Aug. 2020. (Visited on 10/02/2022) (cited on page 56).

[288] Alexander Mordvintsev, Eyvind Niklasson, and Ettore Randazzo. *Particle Lenia and the Energy-Based Formulation*. https://google-research.github.io/self-organising-systems/particle-lenia/. Dec. 2022. (Visited on 03/10/2023) (cited on page 56).

[289] James Bradbury et al. *JAX: Composable Transformations of Python+NumPy Programs*. 2018 (cited on pages 56, 79, 135, 151).

[290] Robert Tjarko Lange. *Evosax: JAX-based Evolution Strategies*. Dec. 2022. DOI: 10.48550/arXiv.2212.04180. (Visited on 01/31/2023) (cited on pages 58, 151).

[291] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization." In: *arXiv preprint arXiv:1412.6980* (2014) (cited on pages 58, 200).

[292] Tim Taylor. *Requirements for Open-Ended Evolution in Natural and Artificial Systems*. July 2015. DOI: 10.48550/arXiv.1507.07403. (Visited on 03/14/2024) (cited on page 61).

[293] Samuel Arbesman. *Emergent Microcosms*. Substack Newsletter. Dec. 2022. (Visited on 02/28/2023) (cited on page 70).

[294] Peter Godfrey-Smith. "Environmental Complexity and the Evolution of Cognition." In: *The Evolution of Intelligence*. Mahwah, NJ, US: Lawrence Erlbaum Associates Publishers, 2002, pp. 223–249 (cited on page 70).

[295] Michael Levin. "Darwin's Agential Materials: Evolutionary Implications of Multiscale Competency in Developmental Biology." In: *Cellular and Molecular Life Sciences* 80.6 (2023), p. 142. DOI: 10.1007/s00018-023-04790-z. (Visited on 03/01/2024) (cited on page 71).

[296] Keith Yuan Patarroyo et al. "AssemblyCA: A Benchmark of Open-Endedness for Discrete Cellular Automata." In: *Second Agent Learning in Open-Endedness Workshop*. Dec. 2023. (Visited on 03/01/2024) (cited on page 71).

[297] Abhishek Sharma et al. "Assembly Theory Explains and Quantifies Selection and Evolution." In: *Nature* 622.7982 (Oct. 2023), pp. 321–328. DOI: 10.1038/s41586-023-06600-9. (Visited on 03/01/2024) (cited on page 71).

[298] Elinor Ostrom. *Governing the commons: The evolution of institutions for collective action*. Cambridge university press, 1990 (cited on page 75).

[299] Garrett Hardin. "The Tragedy of the Commons." In: *Science* 162.3859 (1968), pp. 1243–1248 (cited on pages 75, 81).

[300] Julien Perolat et al. "A multi-agent reinforcement learning model of common-pool resource appropriation." In: *Advances in neural information processing systems* 30 (2017) (cited on pages 78, 81, 92, 93, 105, 188).

[301] Joel Z. Leibo et al. *Multi-Agent Reinforcement Learning in Sequential Social Dilemmas*. Tech. rep. arXiv:1702.03037. arXiv, Feb. 2017 (cited on pages 78, 81, 188).

[302] Eleni Nisioti and Clément Moulin-Frier. "Plasticity and evolvability under environmental variability: the joint role of fitness-based selection and niche-limited competition." In: *Proceedings of the Genetic and Evolutionary Computation Conference*. 2022, pp. 113–121 (cited on pages 78, 81, 153, 189).

[303] Jonathan C. Brant and Kenneth O. Stanley. "Minimal Criterion Coevolution: A New Approach to Open-Ended Search." In: *Proceedings of the Genetic and Evolutionary Computation Conference*. GECCO '17. New York, NY, USA: Association for Computing Machinery, July 2017, pp. 67–74. DOI: 10.1145/3071178.3071186 (cited on pages 78–80, 190).

[304] Kenneth O. Stanley et al. "Designing Neural Networks through Neuroevolution." In: *Nature Machine Intelligence* 1.1 (Jan. 2019), pp. 24–35. DOI: 10.1038/s42256-018-0006-z (cited on pages 78, 80, 83, 190).

[305] Sebastian Risi, Charles E Hughes, and Kenneth O Stanley. "Evolving Plastic Neural Networks with Novelty Search." In: *Adaptive Behavior* 18.6 (Dec. 2010), pp. 470–491. DOI: 10.1177/1059712310379923 (cited on page 80).

[306] Risto Miikkulainen et al. "Multiagent Learning through Neuroevolution." In: *Advances in Computational Intelligence*. Ed. by Jing Liu et al. Vol. 7311. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 24–46. DOI: 10.1007/978-3-642-30687-7_2 (cited on page 80).

[307] Brian G. Woolley and Kenneth O. Stanley. "On the Deleterious Effects of a Priori Objectives on Evolution and Representation." In: *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*. GECCO '11. New York, NY, USA: Association for Computing Machinery, July 2011, pp. 957–964. DOI: 10.1145/2001576.2001707 (cited on page 80).

[308] Abhishek Gupta et al. "Reset-Free Reinforcement Learning via Multi-Task Learning: Learning Dexterous Manipulation Behaviors without Human Intervention." In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. Xi'an, China: IEEE Press, May 2021, pp. 6664–6671. DOI: 10.1109/ICRA48506.2021.9561384 (cited on page 80).

[309] Marco A. Janssen. "Introducing Ecological Dynamics into Common-Pool Resource Experiments." In: *Ecology and Society* 15.2 (2010), art7. DOI: 10.5751/ES-03296-150207 (cited on page 81).

[310] G A Parker. "Scramble in Behaviour and Ecology." In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 355.1403 (Nov. 2000), pp. 1637–1645 (cited on page 81).

[311] Nicolas Anastassacos et al. "Cooperation and Reputation Dynamics with Reinforcement Learning." In: *arXiv:2102.07523 [cs]* (Feb. 2021) (cited on page 81).

[312] Matt Grove. "Evolution and Dispersal under Climatic Instability: A Simple Evolutionary Algorithm." In: *Adaptive Behavior* 22.4 (Aug. 2014), pp. 235–254. DOI: 10.1177/1059712314533573 (cited on pages 81, 189).

[313] Robert Kirk et al. "A Survey of Zero-shot Generalisation in Deep Reinforcement Learning." In: *Journal of Artificial Intelligence Research* 76 (Jan. 2023), pp. 201–264. DOI: 10.1613/jair.1.14174 (cited on page 83).

[314] Vito Volterra. "Fluctuations in the Abundance of a Species considered Mathematically." In: *Nature* 118 (1926), pp. 558–560 (cited on page 85).

[315] Nathanaël Aubert-Kato, Olaf Witkowski, and Takashi Ikegami. "The Hunger Games: Embodied agents evolving foraging strategies on the frugal-greedy spectrum." In: vol. ECAL 2015: the 13th European Conference on Artificial Life. 2015, pp. 357–364. DOI: 10.1162/978-0-262-33027-5-ch065 (cited on page 85).

[316] Frans de Waal. *Are We Smart Enough to Know How Smart Animals Are?* Are We Smart Enough to Know How Smart Animals Are? New York, NY, US: W W Norton & Co, 2016, p. 340 (cited on page 88).

[317] John Gowdy and Lisi Krall. "Agriculture as a major evolutionary transition to human ultrasociality." In: *Journal of Bioeconomics* 16.2 (2014), pp. 179–202 (cited on page 89).

[318] Ted R Schultz and Seán G Brady. "Major evolutionary transitions in ant agriculture." In: *Proceedings of the National Academy of Sciences* 105.14 (2008), pp. 5435–5440 (cited on page 89).

[319] Ulrich G Mueller et al. "The evolution of agriculture in insects." In: *Annu. Rev. Ecol. Evol. Syst.* 36.1 (2005), pp. 563–595 (cited on page 89).

[320] John Schulman et al. *Proximal Policy Optimization Algorithms*. 2017 (cited on pages 93, 110, 135).

[321] Eörs Szathmáry. "Toward major evolutionary transitions theory 2.0." In: *Proceedings of the National Academy of Sciences* 112.33 (2015), pp. 10104–10111 (cited on page 97).

[322] Christopher R Gignoux, Brenna M Henn, and Joanna L Mountain. "Rapid, global demographic expansions after the origins of agriculture." In: *Proceedings of the National Academy of Sciences* 108.15 (2011), pp. 6044–6049 (cited on page 99).

[323] Sanne Nygaard et al. "Reciprocal genomic evolution in the ant–fungus agricultural symbiosis." In: *Nature Communications* 7.1 (2016), p. 12233 (cited on page 99).

[324] Alain Testart et al. "The significance of food storage among hunter-gatherers: Residence patterns, population densities, and social inequalities [and comments and reply]." In: *Current anthropology* 23.5 (1982), pp. 523–537 (cited on page 99).

[325] Andrea Matranga. "The ant and the grasshopper: Seasonality and the invention of agriculture." In: *The Quarterly Journal of Economics* (2024), qjae012 (cited on pages 99, 146).

[326] Slimane Dridi and Laurent Lehmann. "Environmental complexity favors the evolution of learning." In: *Behavioral Ecology* 27.3 (2016), pp. 842–850 (cited on page 101).

[327] Daniel E Berlyne. "Novelty and curiosity as determinants of exploratory behaviour." In: *British journal of psychology* 41.1 (1950), p. 68 (cited on page 101).

[328] Robert W White. "Motivation reconsidered: the concept of competence." In: *Psychological review* 66.5 (1959), p. 297 (cited on page 101).

[329] Daniel E Berlyne. "Curiosity and Exploration: Animals spend much of their time seeking stimuli whose significance raises problems for psychology." In: *Science* 153.3731 (1966), pp. 25–33 (cited on pages 101, 119).

[330] George Loewenstein. "The psychology of curiosity: A review and reinterpretation." In: *Psychological bulletin* 116.1 (1994), p. 75 (cited on page 101).

[331] Celeste Kidd and Benjamin Y Hayden. "The psychology and neuroscience of curiosity." In: *Neuron* 88.3 (2015), pp. 449–460 (cited on page 101).

[332] Satinder Singh et al. "Intrinsically Motivated Reinforcement Learning: An Evolutionary Perspective." In: *IEEE Transactions on Autonomous Mental Development* 2.2 (2010), pp. 70–82. DOI: 10.1109/TAMD.2010.2051031 (cited on page 102).

[333] Ben Norman and Jeff Clune. "First-Explore, then Exploit: Meta-Learning to Solve Hard Exploration-Exploitation Trade-Offs." In: *NeurIPS 2024 Workshop on Open-World Agents* (cited on pages 102, 136).

[334] Evan Z Liu et al. "Decoupling exploration and exploitation for meta-reinforcement learning without sacrifices." In: *International conference on machine learning*. PMLR. 2021, pp. 6925–6935 (cited on pages 102, 136).

[335] Jürgen Schmidhuber. "Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook." PhD thesis. Technische Universität München, 1987 (cited on page 102).

[336] Sven Gronauer and Klaus Diepold. "Multi-agent deep reinforcement learning: a survey." In: *Artificial Intelligence Review* 55.2 (2022), pp. 895–943 (cited on page 102).

[337] Gregory Palmer. *Independent learning approaches: Overcoming multi-agent learning pathologies in team-games*. The University of Liverpool (United Kingdom), 2020 (cited on page 102).

[338] Michael Tomasello and Malinda Carpenter. "Shared intentionality." In: *Developmental science* 10.1 (2007), pp. 121–125 (cited on pages 103, 119, 144).

[339] Liviu Panait and Sean Luke. "Cooperative multi-agent learning: The state of the art." In: *Autonomous agents and multi-agent systems* 11 (2005), pp. 387–434 (cited on pages 105, 106).

[340] Ming Tan. "Multi-agent reinforcement learning: Independent vs. cooperative agents." In: *Proceedings of the tenth international conference on machine learning*. 1993, pp. 330–337 (cited on pages 105, 106).

[341] Yuko Ishiwaka, Takamasa Sato, and Yukinori Kakazu. "An approach to the pursuit problem on a heterogeneous multiagent system using reinforcement learning." In: *Robotics and Autonomous Systems* 43.4 (2003), pp. 245–256 (cited on pages 105, 106).

[342] Christian Schroeder de Witt et al. "Is independent learning all you need in the starcraft multi-agent challenge?" In: *arXiv preprint arXiv:2011.09533* (2020) (cited on pages 105, 106, 110).

[343] Siqi Liu et al. *Emergent Coordination Through Competition*. 2019 (cited on pages 105, 106).

[344] Mikayel Samvelyan et al. "The starcraft multi-agent challenge." In: *arXiv preprint arXiv:1902.04043* (2019) (cited on page 106).

[345] Juergen Schmidhuber, Jieyu Zhao, and MA Wiering. "Simple principles of metalearning." In: *Technical report IDSIA* 69 (1996), pp. 1–23 (cited on page 106).

[346] Tianhe Yu et al. "Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning." In: *Conference on robot learning*. PMLR. 2020, pp. 1094–1100 (cited on page 107).

[347] Anusha Nagabandi et al. "Learning to adapt in dynamic, real-world environments through meta-reinforcement learning." In: *arXiv preprint arXiv:1803.11347* (2018) (cited on page 107).

[348] Emilio Parisotto et al. "Concurrent meta reinforcement learning." In: *arXiv preprint arXiv:1903.02710* (2019) (cited on page 107).

[349] Mikayel Samvelyan et al. "MAESTRO: Open-ended environment design for multi-agent reinforcement learning." In: *arXiv preprint arXiv:2303.03376* (2023) (cited on page 107).

[350] Michael Garcia Ortiz et al. *Simple-Playgrounds*. https://github.com/mgarciaortiz/simple-playgrounds. 2021 (cited on pages 108, 126, 199).

[351] Zahra Rahaie and Hamid Beigy. "Critic learning in multi agent credit assignment problem." In: *Journal of Intelligent & Fuzzy Systems* 30.6 (2016), pp. 3465–3480 (cited on page 111).

[352] Oriol Vinyals et al. *StarCraft II: A New Challenge for Reinforcement Learning*. 2017. DOI: 10.48550/ARXIV.1708.04782. URL: https://arxiv.org/abs/1708.04782 (cited on page 118).

[353] Rose E. Wang et al. "Model-Based Reinforcement Learning for Decentralized Multiagent Rendezvous." In: *arXiv:2003.06906 [cs]* (Nov. 2020) (cited on pages 118, 121).

[354] Nuttapong Chentanez, Andrew Barto, and Satinder Singh. "Intrinsically Motivated Reinforcement Learning." In: *Advances in Neural Information Processing Systems*. Ed. by L. Saul, Y. Weiss, and L. Bottou. Vol. 17. MIT Press, 2004 (cited on page 118).

[355] Alison Gopnik, Andrew N Meltzoff, and Patricia K Kuhl. *The scientist in the crib: Minds, brains, and how children learn*. William Morrow & Co, 1999 (cited on page 119).

[356] Felix Warneken et al. "Young children's planning in a collaborative problem-solving task." In: *Cognitive Development* 31 (2014), pp. 48–58 (cited on page 119).

[357] Kenneth H Rubin, Kathryn S Watson, and Thomas W Jambor. "Free-play behaviors in preschool and kindergarten children." In: *Child development* (1978), pp. 534–536 (cited on page 119).

[358] Katharina Hamann, Felix Warneken, and Michael Tomasello. "Children's developing commitments to joint goals." In: *Child development* 83.1 (2012), pp. 137–145 (cited on page 119).

[359] Jiachen Yang et al. "CM3: Cooperative Multi-goal Multi-stage Multi-agent Reinforcement Learning." In: *International Conference on Learning Representations*. 2020 (cited on pages 119, 120, 123, 124, 127, 201).

[360] L. Kaelbling. "Learning to Achieve Goals." In: *International Joint Conference on Artificial Intelligence*. 1993. (Visited on 04/26/2023) (cited on pages 120, 122).

[361] Richard S. Sutton, Doina Precup, and Satinder Singh. "Between MDPs and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning." In: *Artificial Intelligence* 112.1 (Aug. 1999), pp. 181–211. DOI: 10.1016/S0004-3702(99)00052-1. (Visited on 04/26/2023) (cited on page 120).

[362] George Konidaris and Andrew Barto. "Skill Discovery in Continuous Reinforcement Learning Domains Using Skill Chaining." In: *Advances in Neural Information Processing Systems*. Vol. 22. Curran Associates, Inc., 2009 (cited on page 120).

[363] Marcin Andrychowicz et al. "Hindsight experience replay." In: *Advances in neural information processing systems* 30 (2017) (cited on page 120).

[364] Tom Schaul et al. "Universal Value Function Approximators." In: *Proceedings of the 32nd International Conference on Machine Learning*. PMLR, June 2015, pp. 1312–1320. (Visited on 04/26/2023) (cited on pages 120, 122).

[365] Igor Mordatch and Pieter Abbeel. "Emergence of grounded compositional language in multi-agent populations." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018 (cited on page 120).

[366] Sao Mai Nguyen and Pierre-Yves Oudeyer. "Socially Guided Intrinsic Motivation for Robot Learning of Motor Skills." In: *Autonomous Robots* 36.3 (Mar. 2014), pp. 273–294. DOI: 10.1007/s10514-013-9339-y (cited on page 120).

[367] German I. Parisi et al. "Continual Lifelong Learning with Neural Networks: A Review." In: *Neural Networks* 113 (May 2019), pp. 54–71. DOI: 10.1016/j.neunet.2019.01.012 (cited on page 120).

[368] Joshua Achiam and Shankar Sastry. *Surprise-Based Intrinsic Motivation for Deep Reinforcement Learning*. Mar. 2017. DOI: 10.48550/arXiv.1703.01732 (cited on page 120).

[369] Juergen Schmidhuber. *Driven by Compression Progress: A Simple Principle Explains Essential Aspects of Subjective Beauty, Novelty, Surprise, Interestingness, Attention, Curiosity, Creativity, Art, Science, Music, Jokes*. Apr. 2009. DOI: 10.48550/arXiv.0812.4360 (cited on page 120).

[370] Natasha Jaques et al. "Social influence as intrinsic motivation for multi-agent deep reinforcement learning." In: *International conference on machine learning*. PMLR. 2019, pp. 3040–3049 (cited on page 120).

[371] Sao Mai Nguyen and Pierre-Yves Oudeyer. "Socially guided intrinsic motivation for robot learning of motor skills." In: *Autonomous Robots* 36.3 (2014), pp. 273–294 (cited on page 121).

[372] Jakob Foerster et al. *Counterfactual Multi-Agent Policy Gradients*. Dec. 2017. DOI: 10.48550/arXiv.1705.08926. (Visited on 04/26/2023) (cited on pages 121, 127).

[373] Ryan Lowe et al. "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments." In: *arXiv:1706.02275 [cs]* (Mar. 2020). (Visited on 01/07/2021) (cited on page 121).

[374] Jakob N. Foerster et al. "Learning to Communicate with Deep Multi-Agent Reinforcement Learning." In: *arXiv:1605.06676 [cs]* (May 2016) (cited on page 121).

[375] Angeliki Lazaridou and Marco Baroni. "Emergent Multi-Agent Communication in the Deep Learning Era." In: *arXiv:2006.02419 [cs]* (July 2020) (cited on page 121).

[376] Luc L. Steels. *The Talking Heads experiment*. Computational Models of Language Evolution 1. Berlin: Language Science Press, 2015 (cited on pages 121, 125).

[377] R. Sutton et al. "Horde: A Scalable Real-Time Architecture for Learning Knowledge from Unsupervised Sensorimotor Interaction." In: *Adaptive Agents and Multi-Agent Systems*. May 2011 (cited on page 122).

[378] Vitchyr H. Pong et al. *Skew-Fit: State-Covering Self-Supervised Reinforcement Learning*. Aug. 2020. DOI: 10.48550/arXiv.1903.03698. (Visited on 04/26/2023) (cited on page 122).

[379] Sebastian Blaes et al. *Control What You Can: Intrinsically Motivated Task-Planning Agent*. Jan. 2020. DOI: 10.48550/arXiv.1906.08190. (Visited on 04/26/2023) (cited on page 122).

[380] Ashvin V Nair et al. "Visual Reinforcement Learning with Imagined Goals." In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018 (cited on page 122).

[381] Michael L. Littman. "Markov Games as a Framework for Multi-Agent Reinforcement Learning." In: *Proceedings of the Eleventh International Conference on International Conference on Machine Learning*. ICML'94. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., July 1994, pp. 157–163 (cited on page 123).

[382] Daniel S. Bernstein, Shlomo Zilberstein, and Neil Immerman. *The Complexity of Decentralized Control of Markov Decision Processes*. Jan. 2013. DOI: 10.48550/arXiv.1301.3836 (cited on page 123).

[383] Brian Skyrms. "The Stag Hunt." In: *Proceedings and Addresses of the American Philosophical Association* 75.2 (2001), pp. 31–41. DOI: 10.2307/3218711 (cited on page 127).

[384] Chris Bamford. *Grafter*. https://github.com/GriddlyAI/grafter. 2022 (cited on page 132).

[385] Yoann Lemesle et al. *Emergence of Shared Sensory-motor Graphical Language from Visual Input*. 2022. DOI: 10.48550/ARXIV.2210.06468. URL: https://arxiv.org/abs/2210.06468 (cited on page 132).

[386] Gautier Dagan, Dieuwke Hupkes, and Elia Bruni. "Co-Evolution of Language and Agents in Referential Games." In: *arXiv:2001.03361 [cs]* (Jan. 2021) (cited on page 132).

[387] Clément Moulin-Frier and Pierre-Yves Oudeyer. "Multi-Agent Reinforcement Learning as a Computational Tool for Language Evolution Research: Historical Context and Future Challenges." In: *Challenges and Opportunities for Multi-Agent Reinforcement Learning (COMARL), AAAI Spring Symposium Series, Stanford University, Palo Alto, California, USA*. 2021. (Visited on 04/07/2021) (cited on page 132).

[388] Tristan Karch et al. "Contrastive Multimodal Learning for Emergence of Graphical Sensory-Motor Communication." In: *arXiv preprint arXiv:2210.06468* (2022) (cited on page 133).

[389] Zihang Dai. "Transformer-xl: Attentive language models beyond a fixed-length context." In: *arXiv preprint arXiv:1901.02860* (2019) (cited on page 135).

[390] Alexander Nikulin et al. "XLand-minigrid: Scalable meta-reinforcement learning environments in JAX." In: *arXiv preprint arXiv:2312.12044* (2023) (cited on pages 146, 147, 149).

[391] Russell K Standish. "Open-ended artificial evolution." In: *International Journal of Computational Intelligence and Applications* 3.02 (2003), pp. 167–175 (cited on page 146).

[392] Olivier Sigaud et al. "A definition of open-ended learning problems for goal-conditioned agents." In: *arXiv preprint arXiv:2311.00344* (2023) (cited on pages 146, 157).

[393] Tianchen He et al. "Possible links between extreme oxygen perturbations and the Cambrian radiation of animals." In: *Nature Geoscience* 12.6 (2019), pp. 468–474 (cited on page 146).

[394] Haijun Song et al. "Thresholds of temperature change for mass extinctions." In: *Nature communications* 12.1 (2021), p. 4694 (cited on page 147).

[395] Jane X Wang et al. "Alchemy: A benchmark and analysis toolkit for meta-reinforcement learning agents." In: *arXiv preprint arXiv:2102.02926* (2021) (cited on page 147).

[396] Danijar Hafner. "Benchmarking the Spectrum of Agent Capabilities." In: *arXiv preprint arXiv:2109.06780* (2021) (cited on page 147).

[397] Ruoyao Wang et al. "Scienceworld: Is your agent smarter than a 5th grader?" In: *arXiv preprint arXiv:2203.07540* (2022) (cited on page 147).

[398] Jake Bruce et al. "Genie: Generative interactive environments." In: *Forty-first International Conference on Machine Learning*. 2024 (cited on page 148).

[399] Mengjiao Yang et al. "Learning interactive real-world simulators." In: *arXiv preprint arXiv:2310.06114* (2023) (cited on page 148).

[400] Paul Rendell. "Turing universality of the game of life." In: *Collision-based computing*. Springer, 2002, pp. 513–539 (cited on page 148).

[401] Sam Earle and Julian Togelius. *Autoverse: An Evolvable Game Language for Learning Robust Embodied Agents*. 2024. URL: https://arxiv.org/abs/2407.04221 (cited on page 148).

[402] Martí Sánchez-Fibla, Clément Moulin-Frier, and Ricard Solé. "Cooperative control of environmental extremes by artificial intelligent agents." In: *Journal of the Royal Society Interface* 21.220 (2024), p. 20240344 (cited on page 149).

[403] Mikayel Samvelyan et al. "Minihack the planet: A sandbox for open-ended reinforcement learning research." In: *arXiv preprint arXiv:2109.13202* (2021) (cited on page 149).

[404] Joar Skalse et al. "Defining and characterizing reward gaming." In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 9460–9471 (cited on page 150).

[405] François Chollet. *On the Measure of Intelligence*. arXiv:1911.01547 [cs]. 2019. DOI: 10.48550/arXiv.1911.01547. URL: http://arxiv.org/abs/1911.01547 (cited on pages 150, 151).

[406] Hosung Lee et al. "Arcle: The abstraction and reasoning corpus learning environment for reinforcement learning." In: *arXiv preprint arXiv:2407.20806* (2024) (cited on page 151).

[407] Grgur Kovač, Adrien Laversanne-Finot, and Pierre-Yves Oudeyer. "GRIMGEP: Learning Progress for Robust Goal Sampling in Visual Deep Reinforcement Learning." In: *IEEE Transactions on Cognitive and Developmental Systems* (2022), pp. 1–1. DOI: 10.1109/TCDS.2022.3216911 (cited on pages 151, 205).

[408] Kuno Kim et al. "Active world model learning with progress curiosity." In: *International conference on machine learning*. PMLR. 2020, pp. 5306–5315 (cited on page 151).

[409] Adam Gaier and David Ha. "Weight agnostic neural networks." In: *Advances in neural information processing systems* 32 (2019) (cited on page 152).

[410] Elias Najarro, Shyam Sudhakaran, and Sebastian Risi. "Towards self-assembling artificial neural networks through neural developmental programs." In: *Artificial Life Conference Proceedings 35*. Vol. 2023. 1. MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info ... 2023, p. 80 (cited on page 152).

[411] Herbert Jaeger. "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note." In: *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report* 148.34 (2001), p. 13 (cited on page 152).

[412] Herbert Jaeger. "Echo state network." In: *scholarpedia* 2.9 (2007), p. 2330 (cited on page 152).

[413] Guanzhi Wang et al. "Voyager: An open-ended embodied agent with large language models." In: *arXiv preprint arXiv:2305.16291* (2023) (cited on page 152).

[414] Shengran Hu, Cong Lu, and Jeff Clune. "Automated design of agentic systems." In: *arXiv preprint arXiv:2408.08435* (2024) (cited on page 152).

[415] Rolf Pfeifer and Gabriel Gómez. "Morphological computation–connecting brain, body, and environment." In: *Creating brain-like intelligence: From basic principles to complex intelligent systems* (2009), pp. 66–83 (cited on page 152).

[416] Karl Sims. "Evolving virtual creatures." In: *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*. 2023, pp. 699–706 (cited on page 152).

[417] Josh Bongard. "The utility of evolving simulated robot morphology increases with task complexity for object manipulation." In: *Artificial life* 16.3 (2010), pp. 201–223 (cited on page 152).

[418] Léni K Le Goff, Edgar Buchanan, and Emma Hart. "An Investigation of the Factors Influencing Evolutionary Dynamics in the Joint Evolution of Robot Body and Control." In: *arXiv preprint arXiv:2403.10303* (2024) (cited on page 152).

[419] Fabio Tanaka and Claus Aranha. "Co-evolving morphology and control of soft robots using a single genome." In: *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE. 2022, pp. 1235–1242 (cited on page 152).

[420] Sidney Pontes-Filho et al. "A single neural cellular automaton for body-brain co-evolution." In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 2022, pp. 148–151 (cited on page 152).

[421] Allan Zhao et al. "RoboGrammar: Graph Grammar for Terrain-Optimized Robot Design." In: *ACM Transactions on Graphics (TOG)* 39.6 (2020), pp. 1–16 (cited on page 152).

[422] Iuliia Parfenova. "Mechanisms facilitating decision-making and memory in response to pheromone in budding yeast S. cerevisiae." PhD thesis. ETH Zurich, 2022 (cited on page 153).

[423] Mayalen Etcheverry et al. "Ai-driven automated discovery tools reveal diverse behavioral competencies of biological networks." In: *Elife* 13 (2025), RP92683 (cited on page 153).

[424] David J Earl and Michael W Deem. "Evolvability is a selectable trait." In: *Proceedings of the National Academy of sciences* 101.32 (2004), pp. 11531–11536 (cited on page 153).

[425] Milton L Montero et al. "Meta-Learning an Evolvable Developmental Encoding." In: *arXiv preprint arXiv:2406.09020* (2024) (cited on page 153).

[426] S Wolfram. *Why does biological evolution work? A minimal model for biological evolution and other adaptive processes*. 2024 (cited on page 153).

[427] Nathan Gaylinn and Joshua Bongard. "A Meta-Evolutionary Algorithm for Co-evolving Genotypes and Genotype/Phenotype Maps." In: *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 2024, pp. 467–470 (cited on page 153).

[428] Simon J Hickinbotham, Susan Stepney, and Paulien Hogeweg. "Nothing in evolution makes sense except in the light of parasitism: evolution of complex replication strategies." In: *Royal Society Open Science* 8.8 (2021), p. 210441 (cited on page 155).

[429] Luís F Seoane and Ricard Solé. "How Turing parasites expand the computational landscape of digital life." In: *Physical Review E* 108.4 (2023), p. 044407 (cited on page 155).

[430] Akarsh Kumar et al. "Automating the Search for Artificial Life with Foundation Models." In: *arXiv preprint arXiv:2412.17799* (2024) (cited on page 156).

[431] Mark A Bedau[1], Emile Snyder, and Norman H Packard[1]. "A classification of long-term evolutionary dynamics." In: *Artificial Life: The Proceedings...* (1998), p. 228 (cited on page 157).

[432] Alastair Channon. “Passing the ALife test: Activity statistics classify evolution in Geb as unbounded.” In: *Advances in Artificial Life: 6th European Conference, ECAL 2001 Prague, Czech Republic, September 10–14, 2001 Proceedings 6*. Springer. 2001, pp. 417–426 (cited on page 157).

[433] Susan Stepney. “Modelling and measuring open-endedness.” In: *Artificial Life* 25.1 (2021), p. 9 (cited on page 157).

[434] Susan Stepney and Simon Hickinbotham. “On the open-endedness of detecting open-endedness.” In: *Artificial Life* 30.3 (2024), pp. 390–416 (cited on page 157).

[435] William Gilpin. “Cellular automata as convolutional neural networks.” In: *Physical Review E* 100.3 (2019), p. 032402 (cited on page 170).

[436] Harshitha S. Kotian et al. “Active modulation of surfactant-driven flow instabilities by swarming bacteria.” In: *Phys. Rev. E* 101 (1 2020), p. 012407. DOI: *10.1103/PhysRevE.101.012407* (cited on page 171).

[437] Jonathan C. Brant and Kenneth O. Stanley. “Diversity preservation in minimal criterion coevolution through resource limitation.” In: *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. GECCO ’20. New York, NY, USA: Association for Computing Machinery, June 2020, pp. 58–66. DOI: *10.1145/3377930.3389809*. (Visited on 10/07/2020) (cited on page 190).

[438] Eric Liang et al. “RLlib: Abstractions for Distributed Reinforcement Learning.” In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by Jennifer Dy and Andreas Krause. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 3053–3062 (cited on page 199).

[439] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library.” In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035 (cited on page 199).

[440] Marcin Andrychowicz et al. “What matters for on-policy deep actor-critic methods? a large-scale study.” In: *International conference on learning representations*. 2020 (cited on page 200).

[441] John Schulman et al. “High-dimensional continuous control using generalized advantage estimation.” In: *arXiv preprint arXiv:1506.02438* (2015) (cited on page 200).

[442] Caroline Claus and Craig Boutilier. “The dynamics of reinforcement learning in cooperative multiagent systems.” In: *AAAI/IAAI* 1998.746-752 (1998), p. 2 (cited on page 201).