Exploring Self-Organization in Complex Biological Systems: From Neural Networks to Evolutionary Dynamics

Dissertation
der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von
Sina Khajehabdollahi
aus
Teheran, Iran

Tübingen
2024

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	22.01.2025
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatterin:	Jun.-Prof. Dr. Anna Levina
2. Berichterstatter:	Prof. Dr. Martin Butz

Acknowledgements

During my time working on my PhD and living in Tübingen I had the fortune of meeting what felt like a whole lifetime worth of people and characters. So much of my existence in this town was a result of a maelstrom of human connections, serendipity and good luck. From the onset, when I originally met Georg as a raised hand in the crowd in the 2018 ALIFE conference in Tokyo, to my introductions and discussion with Anna as a scientist working on self-organized criticality, to last minute conference friends that would one day introduce me to my new home for 5 years in Tübingen, it has always been through these serendipitous personal connections that life flowed forward. Now this time is coming to an end, and I find myself contemplative for how I got here and where I should go next. In such moments, it helps to start with gratitude.

I will always be grateful for my parents, for if not for their unconditional love and encouragement throughout my life, it would be hard to imagine being half the person I am today. To my supervisors Anna Levina and Georg Martius, thank you for humouring this endeavour in the first place, for being patient when things were slow, and encouraging when things were rough. I hope our work together was at the very least amusing, if not exactly ground breaking. In the same vein, I must thank my colleagues turned friends Manos Giannakakis, Roxana Zeraati, Oleg Vinogradov, Victor Buendía, and Tim Schäfer for being kind, for being helpful, for the laughs, for the coffee breaks and the collective existentialism, you all made this academic struggle funnier than it had any business being.

To the friends I met out in the wild, at random bonfires or parties or through the n^{th} mutual friends we inevitably have in this small town, I owe my sanity and happiness to my dear friends Laura Neville and Vanessa Pahl who made the bad times seem OK and the good times phenomenal, thank you for the beautiful memories we had, though I expect to make more. There are not many friends one finds where a single look can carry a conversation, and for that I'm grateful to my close friend Mohammad Bashiri, a kindred spirit. I'll miss the late nights working out an idea followed by an early morning with a Franzbrötchen and a coffee. There are far too many good friends to list each one, as well as a few that I didn't manage to keep that also deserve to be mentioned. But for the sake of brevity, thank you for being in my life, even if for a moment, it no doubt coloured the whole experience in.

Abstract

Collective systems in nature and biology exhibit remarkable self-organizing capabilities, characterized by complex interactions across vast scales in both time and space. Over billions of years, these systems have proliferated and given rise to the immense diversity of life and ecology found on Earth and are now one of the most dominant properties of our planet. The success and proliferation of life on Earth is a testament to how powerful self-organizing processes can be, yet understanding even a subset of these complex living systems is an extremely challenging problem. However, living systems, and even some abiotic complex systems, share many similarities. This raises the questions as to whether there exists universal principles that underpin the diversity of complex systems observed on Earth.

The criticality hypothesis, which views collective systems through the lens of statistical physics, seeks to determine the existence of such universal properties. Originally rooted in the study of magnetism, the properties of critical phenomena, which are usually detected via power-law statistics, have also been found in a variety of biological systems. These include gene-regulatory networks, evolutionary dynamics, animal swarms, or neural activity, suggesting the propensity for living systems to operate near a critical point. These observations suggest that life and complexity may also be critical phenomena, where a growing body of work is now showing how optimal computational properties emerge near critical points. However, a gap remains between theoretical and empirical work on criticality and its application, where this dissertation aims to explore and simulate complex systems and test their relation to criticality.

In three of the publications presented, we find that systems near critical regimes offer advantages in aesthetics, evolutionary dynamics, and neural network optimality. For aesthetics, abstract images with slowly decaying autocorrelations are perceived as more pleasing. In evolutionary dynamics, we demonstrate that embodied Ising neural agents evolve faster near the critical regime but converge to sub-critical states unless task complexity increases. For neural network optimality, we show that the best-performing models on long-memory tasks have more slowly decaying autocorrelations, linked to a closeness to a critical state, and that this performance is strongly influenced by the learning curriculum used.

The remaining two publications explore adaptivity and self-organization. One paper presents methods to simulate large-scale heterogeneous cellular automata (CA), demonstrated on a plastic spiking neural network and an adaptive Ising model. For the Ising model, we show that robust self-organized criticality can be implemented by using heterogeneous and adaptive CA rules. The other paper on neural plasticity in embodied agents shows that the optimality of plasticity rules depends on environmental and task conditions, with rule specificity emerging under certain constraints.

In summary, this dissertation examines complex systems to relate function and performance in applied settings to theoretical principles. While universal properties account for much of their behavior, they are modified by specific constraints, suggesting these properties are useful initial conditions for adaptive systems, which must undergo further optimization in practical applications.

Keywords

complex systems, adaptivity, self-organization, criticality, evolutionary algorithms, optimization, cellular automata, neural networks, plasticity

Kurzfassung

Kollektive Systeme in der Natur und Biologie zeigen bemerkenswerte selbstorganisierende Fähigkeiten, die durch komplexe Wechselwirkungen über große Zeit- und Raumskalen gekennzeichnet sind. Über Milliarden von Jahren haben sich diese Systeme verbreitet und zur immensen Vielfalt des Lebens und der Ökologie auf der Erde beigetragen und sind heute eine der dominierenden Eigenschaften unseres Planeten. Der Erfolg und die Verbreitung des Lebens auf der Erde sind ein Zeugnis für die Macht selbstorganisierender Prozesse, doch selbst ein Teilverständnis dieser komplexen lebenden Systeme stellt eine äußerst schwierige Herausforderung dar. Allerdings teilen lebende Systeme und einige abiotische komplexe Systeme viele Ähnlichkeiten. Dies wirft die Frage auf, ob es universelle Prinzipien gibt, die der Vielfalt der auf der Erde beobachteten komplexen Systeme zugrunde liegen.

Die Kritikalitätshypothese, die kollektive Systeme durch die Linse der statistischen Physik betrachtet, versucht die Existenz solcher universellen Eigenschaften zu bestimmen. Ursprünglich in der Magnetismusforschung verwurzelt, wurden die Eigenschaften kritischer Phänomene, die üblicherweise durch Potenzgesetz-Statistiken erkannt werden, auch in einer Vielzahl biologischer Systeme gefunden. Dazu gehören genregulatorische Netzwerke, evolutionäre Dynamiken, Tierschwärme oder neuronale Aktivitäten, was auf die Neigung lebender Systeme hinweist, in der Nähe eines kritischen Punktes zu operieren. Diese Beobachtungen legen nahe, dass Leben und Komplexität ebenfalls kritische Phänomene sein könnten, wobei eine wachsende Anzahl von Arbeiten nun zeigt, wie optimale Rechenfähigkeiten in der Nähe kritischer Punkte entstehen. Es besteht jedoch eine Lücke zwischen theoretischer und empirischer Arbeit zur Kritikalität und ihrer Anwendung, die diese Dissertation durch die Erforschung und Simulation komplexer Systeme und die Überprüfung ihrer Beziehung zur Kritikalität zu schließen versucht.

In drei der präsentierten Publikationen finden wir, dass Systeme nahe kritischen Regimen Vorteile in Bezug auf Ästhetik, evolutionäre Dynamiken und neuronale Netzwerk-Optimalität bieten. Für die Ästhetik werden abstrakte Bilder mit langsam abklingenden Autokorrelationen als angenehmer wahrgenommen. In den evolutionären Dynamiken zeigen wir, dass verkörperte Ising-Neuronenagenten in der Nähe des kritischen Regimes schneller evolvieren, aber zu subkritischen Zuständen konvergieren, es sei denn, die Aufgabekomplexität nimmt zu. Für die neuronale Netzwerk-Optimalität zeigen wir, dass die leistungsstärksten Modelle bei Langzeitgedächtnisaufgaben langsam abklingende Autokorrelationen aufweisen, was auf eine Nähe zu einem kritischen Zustand hinweist, und dass diese Leistung stark vom verwendeten Lernlehrplan beeinflusst wird.

Die verbleibenden zwei Publikationen erforschen Adaptivität und Selbstorganisation. Ein Papier stellt Methoden zur Simulation großskaliger heterogener zellulärer Automaten (CA) vor, demonstriert an einem plastischen spikenden neuronalen Netzwerk und einem adaptiven Ising-Modell. Für das Ising-Modell zeigen wir, dass eine robuste selbstorganisierte Kritikalität durch die Verwendung heterogener und adaptiver CA-Regeln implementiert werden kann. Das andere Papier zur neuronalen Plastizität in verkörperten Agenten zeigt, dass die Optimalität der Plastizitätsregeln von den Umwelt- und Aufgabenbedingungen abhängt, wobei unter bestimmten Einschränkungen Regel-Spezifizität entsteht.

Zusammenfassend untersucht diese Dissertation komplexe Systeme, um Funktion und Leistung in angewandten Kontexten mit theoretischen Prinzipien zu verknüpfen. Während universelle Eigenschaften einen Großteil ihres Verhaltens erklären, werden sie durch spezifische Einschränkungen modifiziert, was darauf hindeutet, dass diese Eigenschaften nützliche Ausgangsbedingungen für adaptive Systeme sind, die in praktischen Anwendungen weiter optimiert werden müssen.

Contents

List of Figures

1 Introduction and Motivation

1.1 The Emergence of Complexity in Nature

Some of the most interesting phenomena in the natural world tend to fall into the category of complex systems, which will be the central theme of the projects presented within this dissertation. Complex systems, to put it broadly, are systems that are composed of many different interacting parts whose behaviour is considerably more complicated than its constituent parts. These are systems in which the details cannot easily be ignored and the interactions of these distinct components are often strongly nonlinear. As opposed to linear systems which are more readily solvable, nonlinear systems tend to exhibit chaotic behaviours making them difficult to study at scale. Furthermore, complex systems can also sometimes give rise to **emergent** behaviours that are usually not easily predicted from the properties of the smaller components. There are many examples of this in nature, not least of all the emergence of biology, life, intelligent behaviour, or consciousness[1].

Def.: Emergence

A phenomena that occurs when a macroscopic system has properties its microscopic constituents do not have [1]. Complex properties can emerge from a collective of simple components, e.g. fractal growth of snowflakes.

In this dissertation, we explore a set of mathematical models of complex collective systems focusing on the relationships between their functional utility in applied settings, their microscopic details, and universal properties that they may share.

The models are often (but not always) studied in relation to **the criticality hypothesis**. The criticality hypothesis is a candidate theory of complex systems which claims that the optimal state of for information processing and performance is poised at a critical point where a second order phase transition occurs. The relationship between performance in the biological/evolutionary sense and performance in the information-theoretic sense (which occurs near critical points) is still poorly understood and remains to be verified for many real-world applications. We therefore study if such a universal optimal state exists, if they are indeed related to criticality, and if such states would be natural attractors for evolutionary/adaptive systems or if they are obfuscated by local minima or rugged fitness landscapes. We also explore other models of complexity that are not directly related to the criticality hypothesis but still exhibit similar statistical properties. The models in study are as follows:

- **Compositional Pattern-Producing Networks (CPPN):** a type of neural network that can produce textures by mapping a set of coordinates to a vector output. We evaluate the aesthetics of the generated images and analyzed them using autocorrelation functions.
- **Evolving Ising-Neural Agent:** an embodied neural network with Ising interactions subject to an evolutionary algorithm. We test the hypothesis that criticality is an evolutionary attractor.
- **Adaptive Neural Cellular Automata:** a fast implementation of heterogeneous CA allowing for adaptive dynamics and self-organized criticality, enabling fast prototyping and real-time interactive simulations.

Def.: Criticality

A phenomena characterized by sensitivity to perturbations, divergent correlation-lengths, fractal behaviour, etc. Occurs for certain phase-transitions that typically separate an *ordered* phase from a *disordered* phase. Explained in more detail in Section 2.1 – The Criticality Hypothesis and The Edge of Chaos and Section 3.2 – The Ising Model, Phase Transitions, and Critical Points.

[1] How does single cellular life give rise to multi-cellular life? How do schools of fish or flocks of birds coordinate their collective motion [2–5]? How easily can we predict chemistry from the laws of physics [6]? How does qualia (the phenomena of experience) arise from biology? These emergent phenomena are not readily explained by studying the components in a reductionist way.

- **Plastic Neural Agents:** neural networks with linear reward-modulated plasticity rules. Evolved to solve an embodied foraging task, where we study the emergent rules after optimization. We test how topology and task structure affect the evolved plasticity rules.
- **Recurrent Neural Networks (RNN):** trained via different curriculum learning methods and back-propagation on memory tasks. We study mechanisms for the emergence of long timescales.

1.2 Collective Systems and Self-Organization

What connects the different projects and models studied in this dissertation is that they are all models of collective systems which, under ideal circumstances, can be optimized or self-organized in order to exhibit some desired behaviour or function.

A collective system is simply a large number of very similar (often identical)[2] components. An ocean is composed of $\sim 10^{44}$ identical molecules of water, a neural network is made of billions of relatively similar neurons, and a city is composed of millions of semi-individual humans.

The observation that collective systems undergo spontaneous reorganization is usually quite an intuitive experience for us as humans. From the observation that water can be liquid, solid or gas, the metamorphosis of a caterpillar to a butterfly, or even societal reorganizations, all these phenomena are examples of systems whose fundamental components are (mostly) conserved (the molecules, the biomatter, the humans) but through their reorganization have changed their function or properties (a phase transition). Furthermore, it has been observed that collective systems in a broad variety of domains and scales exhibit statistical properties (*e.g.* 1/f noise, power-laws[3]) that exhibit signatures of spontaneous self-organized criticality [7–12].

Practically and historically, models of collective systems aiming to understand these types of phase transitions need to be relatively simple in order to be solvable. The Ising Model, which in many cases is equivalent to a **maximum-entropy model** for complex systems [13, 15–17] (and which will be introduced in more detail in Section 3.2 – The Ising Model, Phase Transitions, and Critical Points), has exact solutions only under very specific conditions that lend enough symmetry/simplicity to be mathematically tractable. Even so, the discovery of critical phase transitions in the 2D Ising model and the universality of such transitions across different substrates has presented us with a compelling discovery. That is, that near these critical points, many disparate systems converge towards the same universal statistical properties and behaviours, and furthermore, these properties are often exploitable in the context of living, energetic, and informational beings.

To give an intuitive idea for why these systems may be evolutionarily attractive, one can imagine two extremes of a dynamical system: one which rapidly converges to a fixed point from any initial condition or perturbation, versus one which either behaves completely randomly or is so sensitive to perturbations such that it never returns to a previous state. On the one hand, sustaining life requires homeostasis and so

Def.: Maximum Entropy Model

A statistical model that estimates probability distributions by maximizing entropy subject to given constraints, ensuring the least biased estimate on the known information. For example, if the only constraints are the mean activity and pair-wise correlations of a set of variables, the maximum-entropy model is equivalent to an Ising model [13, 14].

[2] Reality is a bit messier where collective systems can get quite heterogeneous. Sometimes these differences average out to be irrelevant, other times they can make the behaviour of the system quite complex. It is a non-trivial problem to distinguish between the two cases.

[3] Mathematically, power-laws have scale-invariant properties and tend to show up in phenomena that are self-similar or scale-invariant. They are therefore used as signatures of critical phenomena and universality, though one must be careful because scale-invariance is not the only way to generate power-law statistics.

having stability to perturbations is vital for survival. On the other hand, life requires adaptability in the face of changing circumstances or environments (*e.g.* seasonal changes, the precession of Earth, CO2 cycles, continental drift, meteors, etc.). It is therefore crucial that for a living system to persist, it must be able to be structured and stable as well as explorative, adaptive, and able to modify its behaviour in reaction to changes in its world. A dynamical system that can easily navigate between these different regimes of behaviour might therefore be more capable to survive in a structured but ever-changing world.

Unfortunately, models of criticality are often forced to be relatively simple to be tractably solved analytically. These idealized settings for which these systems are solvable are often far removed from the rougher, more heterogeneous edges of reality, are rarely embedded in real-world tasks or environments and are too simple to relate practically to applied settings. Thus, there remains a massive gap between the foundational research on collective systems that has risen out of statistical mechanics, and the more practical and functional side of collective systems that has risen out of the fields of machine learning and deep learning. One of the goals of this dissertation is to bridge this gap, and study collective systems of *optimized* or *functional*[4] systems to better understand how natural selection pressures or learning constrain or determine the emergent dynamical properties of collective systems.

1.3 Computational Neuroscience and Machine Learning

It is precisely at the intersection of computational neuroscience and machine learning where we can search for a bridge between these idealized systems and functional ones. Machine learning models (and more specifically deep neural networks), were initially directly inspired by their biological counterpart, neural networks in the brain and intelligent behaviour. Unfortunately, the practicalities of the implementations of these neural network models *in silico* very quickly forces practitioners to adhere[5] to hardware limitations in order to compute these models efficiently. Nowadays, our best models of intelligent behaviour (*e.g.* vision models or large language models) have only a limited resemblance to their biological analogues.

There are practical engineering reasons why deep neural networks are different than their biological counterpart. For example, most State-of-the-Art (SOTA) models don't spike the way real neurons do, and that's usually related to the lack of differentiability of spiking models, their expensive simulation times compared to matrix multiplication, and also the design of modern computer chips that can't take advantage of the energetic benefits of spiking models (although these points are slowly changing with the continued development of neuromorphic computers). Furthermore, difficulties with credit-assignment or backpropogation-through-time when training on temporal tasks and the universal problem of exploding/vanishing gradients forces very specific mathematical (and non-biological) pipelines to ensure well-behaving gradients. Each of these problems exposes a hurdle in our ability to emulate what nature

[4] **Optimized** or **functional** models in this dissertation are models that are subject to optimization algorithms such as gradient-descent via backpropagation or evolutionary algorithms in order to accomplish a specific task or minimize/-maximize an loss/objective function.

[5] Due to the hardware capabilities and the success of modern GPUs, the best way to get the most bang for your buck out of a computer is to convert your problem into a series of matrix multiplications or to vectorize the computation. Therefore, most state-of-the-art algorithms leverage highly optimized MatMul operations instead of directly modeling the processes as they occur in the brain with spiking neurons. The necessity of differentiating these operations for back-propagation algorithms also strongly constrains the types of operations and models used in these optimization settings *in silico*.

does, but also allows us an opportunity to wonder how these problems are dealt with in nature. As our electrical and compute demands grow exponentially, we are evermore appreciative of how nature has solved these problems with sparsely connected, spiking neural networks and energy efficient architectures. As the memory costs and differentiation constraints of highly nonlinear models have grown, we have begun to appreciate more how nature has worked around this through various plasticity mechanisms in the brain.

It is therefore clear to me that one of the most interesting frontiers for understanding and creating intelligent collective systems is in the intersection of computational neuroscience and machine learning where a combination of foundational and engineering goals are intertwined and studied in context of one another.

1.4 Contributions

In Chapter 2 – General Background, I give introductions to some of the foundational and recurring topics in my publications. Starting from a high-level introduction, an overview of the criticality hypothesis and its implications for our understanding of the brain and complex systems is given in Section 2.1 – The Criticality Hypothesis and The Edge of Chaos. I introduce the notion of Self-Organized Criticality in Section 2.2 – Self-Organized Criticality (SoC) to motivate the interest behind adaptive and dynamic systems and the capacity for spontaneous organization. I also briefly discuss the topic of the perception of aesthetics and beauty in Section 2.3 – Beauty and Aesthetic Perception to motivate its relationship with learning, adaptability and ultimately, criticality. In Chapter 3 – Models, I introduce a number of candidate models of collective and adaptive systems that I utilize in my publications that allowed us to study the emergence of complexity, self-organization, and functional performance.

The results of my publications as well as individual author contributions are summarized in Chapter 4 – Publications. The publications are attached in the Appendix in Chapter A – Publications and explore the themes of complexity, aesthetics and correlation statistics (Section 4.1 – Assessing Aesthetics of Generated Abstract Images Using Correlation Structure), evolutionary optimality (Section 4.2 – When to be critical? Performance and evolvability in different regimes of neural Ising agents), self-organized criticality in heterogeneous cellular automata (Section 4.3 – Locally adaptive cellular automata for goal-oriented self-organization), plasticity and optimality (Section 4.4 – Environmental variability and network structure determine the optimal plasticity mechanisms in embodied agents), and lastly, the emergence of long timescales in neural networks (Section 4.5 – Emergent mechanisms for long timescales depend on training curriculum and affect performance in memory tasks).

General Background

General Background 2

2.1 The Criticality Hypothesis and The Edge of Chaos

> *One of the great scientific challenges of our time is to understand how the brain processes information. Current work suggests that individual neurons are relatively modest in their computational power, but that networks of neurons can collectively perform extremely complex operations. Any system that sheds light on emergent properties may therefore potentially advance our understanding of neural function.*
>
> Beggs, 2008 [18]

The criticality hypothesis (or **the critical brain hypothesis**) suggests that adaptability and computational power are optimal at the 'edge of chaos' [19–23] or near a critical point at the onset of a second-order phase transition between an ordered and a disordered state. Near this critical point, properties such as susceptibility, dynamic range, and correlation length are maximized, indicating a heightened sensitivity to stimuli and the ability to process information efficiently [18, 24–29]. A critical point is typically described as the threshold separating an ordered phase from a disordered phase, where the system's organization undergoes a dramatic transformation (Figure 2.1). The terms *ordered* and *disordered* refer to the macroscopic structure of a system (e.g., ice crystals exhibit order and lower entropy compared to steam, which is more disordered, turbulent, and has higher entropy).

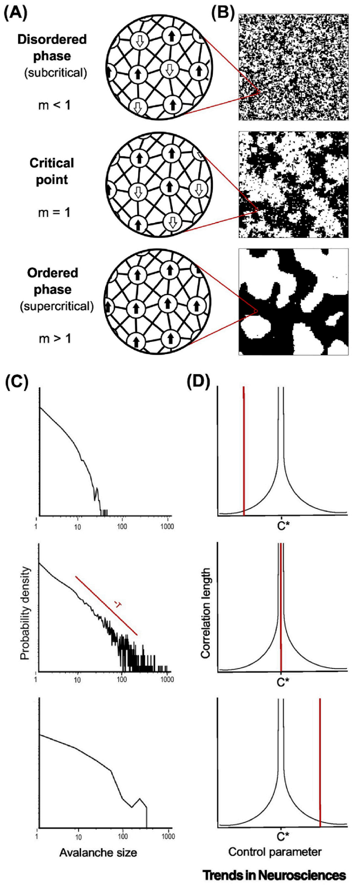


Figure 2.1: Ising-spin statistics **(A, B)**, and neural avalanche statistics **(C)** are related phenomena when analyzed **(D)** in the framework of critical systems. Figure taken from [30].

Key Idea: The Criticality Hypothesis

The central concept of this hypothesis is that systems poised near critical points, at the edge of these transitional states, achieve an optimal balance between opposing dynamical regimes to leverage the benefits of both phases. These systems exhibit both stability (e.g. predictable, structured, with strong attractors ensuring convergent properties) and adaptability (*e.g.*, unstable/meta-stable, capable of fluctuations, able to navigate between attractors when appropriately perturbed). Importantly, these properties arise through collective interactions within the system, emerging even when individual components do not exhibit such characteristics. This emergent behavior enables efficient information propagation and integration over multiple timescales, as well as sensitivity to perturbations across multiple scales—properties considered essential for life and intelligence.

The criticality hypothesis has the potential to transform our understanding of all complex systems, not just the brain. This is partly due to the **universality** [31–34] of phase transitions, where systems with entirely different microscopic details ultimately share the same asymptotic behaviors near their critical points. This universality means that other systems, which are not necessarily similar to the brain at a microscopic level[1], could exhibit similar macroscopic behavior if they belong to the same universality class. If near-critical systems are indeed necessary for the emergence of life and intelligence, then the universality of phase transitions indicates that there are numerous microscopic models capable of exhibiting these properties.

The broader implications of universality extend to how we design, maintain, and diagnose our own complex systems, whether in clinical contexts [2] [44–47], artificial intelligence, or even more broadly in artificial life [48, 49] or life as it *could* be. The diversity of life on Earth demonstrates that there are numerous viable 'solutions' for life across multiple scales, many of which are subject to similar natural selection pressures (*e.g.*, finite resources, changing environments, competition with other organisms, entropic decay, etc.). This degeneracy in the solution space suggests that there are universal properties shared among all these viable solutions. In mathematical terms, can we describe broad categories of dynamical systems that serve as candidate solutions to life? To this end, one of the key motivations of the projects presented in this dissertation has been to verify if indeed systems near criticality exhibit these kinds of universally optimal characteristics, not just in theory, but in settings that are applied, embodied, or optimized for specific tasks.

2.2 Self-Organized Criticality (SoC)

> *In the realm of biology, it is known that life on earth is based on the DNA double helix. But even though we understand perfectly the laws governing the interaction of atoms, we cannot directly extrapolate these laws to explain the beginning of life, or the auto-catalysis of complex molecular networks, or why we have brains that can contemplate the world around us. Due to the overwhelming unlikeliness of random events leading to complex systems like ourselves, it seems as if an organizing agent or "God" must be invoked who puts the building blocks together.*
>
> Paczuski and Bak, 1999 [50]

A dynamical system that naturally evolves to a critical state without precise parameter tuning is said to exhibit **self-organized criticality (SoC)** [7]. SoC was first hypothesized to explain the ubiquitous presence of power-laws in natural and artificial systems. These systems are intriguing because, in contrast to finely-tuned random models, they achieve criticality through their inherent dynamics.

In models like the 2D Ising model, critical states occur under very specific conditions that are difficult to compute [3]. Furthermore, they are often conservative, non-driven, equilibrium systems with large separations of timescales. However, many systems considered to exhibit critical properties, such as earthquakes and neuronal avalanches [8–12], are driven, out-of-equilibrium systems with numerous variables. These should, in theory, be difficult to fine-tune, posing the question: "*Why do*

[1] E.g. a computer, a collective of humans, an ensemble of pigeons detecting cancer [35]...

[2] For example in diagnosing and discriminating between altered brain states that may or may not be pathological (*e.g.* paralyzed, locked-in, or vegetative patients) [36–43].

[3] This challenge has attracted considerable mathematical attention [51, 52], with exact solutions to more complex, less symmetric and more heterogeneous models remaining elusive except through numerical methods.

so many complex, heterogeneous, noisy systems exhibit critical properties? Are there contexts where the critical state is a natural attractor?"

Indeed, systems with critical points as attractors do exist, as evidenced by sandpile models (Figure 2.2), forest-fire models, and earthquake models [12, 54–57]. While these models prove the existence of such phenomena, they do not confirm their occurrence in practical settings or biological systems. This brings us back to the criticality hypothesis (Section 2.1 – The Criticality Hypothesis and The Edge of Chaos). If this hypothesis is true, it would suggest that non-critical systems should evolve towards criticality under performance-related selection pressures.

Thus, the question extends to: *Can evolutionary dynamics induce SoC in non-critical systems?* To explore this, this dissertation analyzes systems optimized through a variety of optimization algorithms (such as evolutionary algorithms or gradient descent via backpropagation), as well as some randomly sampled systems, to estimate their distance to criticality. This analysis aims to elucidate the relationship between performance, optimality, and criticality.

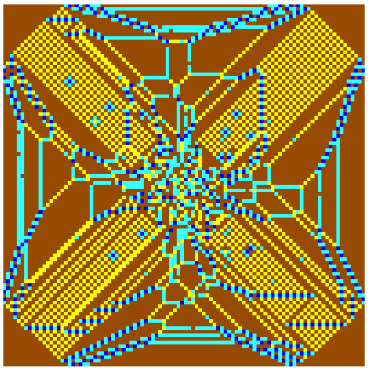


Figure 2.2: Abelian sandpile simulation, demonstrating how self-organizing systems can give rise to both structure and complexity. Figure generated from [53].

2.3 Beauty and Aesthetic Perception

> *In Birkhoff's aesthetics [58] the role of order was to perceptually reward the effort of focusing attention on something complex. He assumed that there exist elements of order such as symmetry, rhythm, repetition, contrast, etc. which psychologically cause a positive tone of feeling, and also elements that cause negative tones, such as ambiguity or undue repetition. To Moles [59], the concept of order was represented by redundancy, which represents a perceiver's a priori knowledge of a received stimulus and keeps complexity down to an interesting or aesthetically pleasant level. More precisely he related order to the degree of predictability and internal coherence, expressed by the concept of mean autocorrelation.*
>
> Neumann et al., 2005 [60]

The perception of beauty and aesthetics is a challenging subject to broach scientifically, as its boundaries and definitions often feel ethereal and have, at various points in history, been attributed to the divine. The subjectivity of beauty and the diversity of aesthetic experiences across different times, people, and cultures further complicate the matter. However, universal features in naturalistic images as well as art and man-made images hints at a underlying universal properties [61–63].This tension between the universality of beauty and its subjectivity has not yet been reconciled into a unified *theory of beauty*. Nevertheless, substantial progress has been made in our understanding and conceptualizations of aesthetics over the centuries.

In the 18th century, the work of Immanuel Kant in his *Critique of Judgment* [64] explicitly bound these apparent contradictions together, arguing that aesthetic judgment is a unique kind of judgement that is both subjective and universal, arising from the pleasure between imagination and understanding. He posited that beauty is not a property of objects themselves but a result of the mind's ability to find a sense of order and purpose in sensory experiences.

In the early 20th century, mathematician George David Birkhoff further developed the study of aesthetics in his 1933 book Aesthetic Measure [58]. Birkhoff proposed that the aesthetic value of an object could be quantified as a ratio of order to complexity, suggesting that beauty arises from a balance between these two elements. His work laid the groundwork for a more analytical approach to aesthetics, introducing mathematical rigor into the traditionally philosophical topic.

Following Birkhoff, various scholars contributed to expanding the idea of quantifying beauty. In the latter half of the 20th century, researchers like Abraham Moles [59] and Max Bense [66], inspired by the formulations of information theory by Claude Shannon [67], integrated information theory with aesthetics. They suggested that the aesthetic experience is tied to the information content and entropy of an artwork. Further work by Jürgen Schmidhuber in the 90s and early 2000s proposed that beauty is related to the observer's ability to compress information [68, 69], linking the perception of beauty to curiosity, exploration and learning. More recently, a theory of aesthetic value has been proposed that attempts to integrate and account for patterns proposed by previous theories [70].

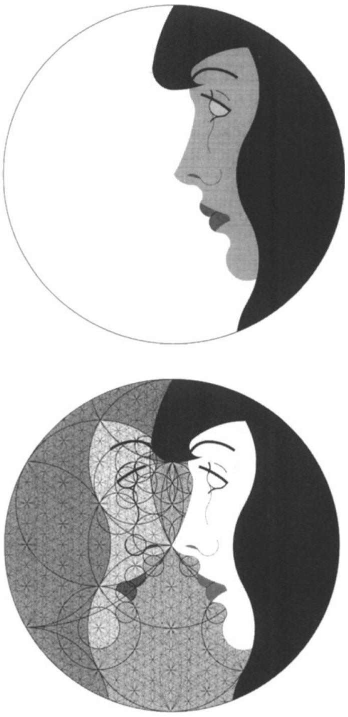


Figure 2.3: An illustration of the low Kolmogorov complexity algorithm that generated the top image. Schmidhuber argues that in many artistic styles, capturing the "essence" of a complex subject in as simple or structured of a way is aesthetically pleasing. He introduces his low-complexity art style as an algorithmic means to this end. Figure taken from [65].

Motivated by these approaches, our publication in Section 4.1 – Assessing Aesthetics of Generated Abstract Images Using Correlation Structure explores this tension between order and disorder in aesthetic perception by relating it to the phenomena of criticality which shares a similar tension. We investigate whether statistical properties, such as slower decaying autocorrelation functions, can predict the aesthetic appeal of images and whether universal statistical properties are present in the perception of aesthetic visual imagery.

Models

Models 3

This chapter introduces the various models utilized in my publications. Each model is initially presented within its broader context and motivation, followed by a detailed technical description.

The sections are related to the following publications:

- Section 3.1 is relevant for the publication Section 4.1 – Assessing Aesthetics of Generated Abstract Images Using Correlation Structure.
- Section 3.2 is relevant for the publication Section 4.2 – When to be critical? Performance and evolvability in different regimes of neural Ising agents.
- Section 3.3 is relevant for the publication Section 4.3 – Locally adaptive cellular automata for goal-oriented self-organization.
- Section 3.4 is relevant for the publications Section 4.3 – Locally adaptive cellular automata for goal-oriented self-organization and Section 4.4 – Environmental variability and network structure determine the optimal plasticity mechanisms in embodied agents.

3.1 Compositional Pattern-Producing Networks (CPPN)

How do $\sim 10^5$ genes encode for $\sim 10^{12}$ interconnecting components, including complex systems such as the human brain? Compositional Pattern-Producing Networks were originally devised as an abstraction of development by creating a mapping between genotype and phenotype that does not require local interactions or temporal unfolding [71]. By arguing that local processes that requires temporal unfolding can be equivalently described by a *functional* description that simply outputs the resulting final state, the authors hypothesize that models like CPPNs can more compactly and efficiently encode complex developmental processes. These insights have have been utilized in the construction of hypernetworks, which are networks that describe other, usually larger, networks as a soft form of weight-sharing [72–74].

CPPNs are a type of artificial neural network which take as input a set of coordinates and output an n-dimensional vector. Since the output of the network is only a function of the coordinates, different coordinates from the same coordinate system can be computed in parallel as they are independent. Due to this simple property that make them easily parallelizable, they are readily adapted to image generation with GPUs and have been found to produce a variety of aesthetic art [75–78]. We use random CPPNs as image generators in the publication Section A.1 – Assessing Aesthetics of Generated Abstract Images Using Correlation Structure. In the introduction below we explore some of the properties of CPPNs such as how the coordinate systems and architecture affect the image output.

3.1.1 Model

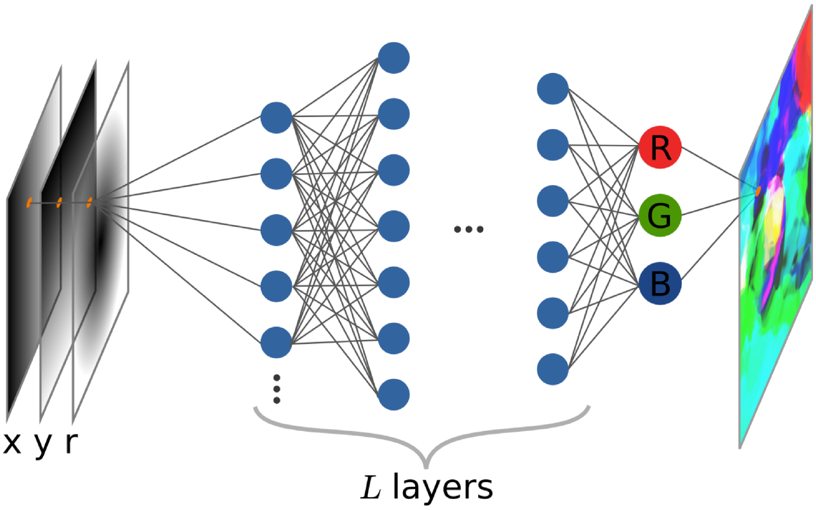


Figure 3.1: Simplified architecture of an image generating CPPN. Taking as input a vector of coordinates, a CPPN composed of simple feed-forward layers outputs a vector that is interpreted as RGB values. Each point in the coordinate space can be computed in parallel as they are independent of each other. The interaction between the structure of the coordinate space, and the architecture of the CPPN produce a variety of complex and beautiful patterns.

Our implementation of the CPPN is defined as follows:

i. Let $\mathbf{x}$ be the input coordinate vector for the CPPN. This can be an n-dimensional vector but for visualization we can use a 2D Cartesian coordinate system that contains (x, y) values (Figure 3.2).

ii. Typically, this spatially defined coordinate is paired with a global, latent vector $\mathbf{z} \in \mathbb{R}^{d_z}$, where d_z is the dimensionality of $\mathbf{z}$ which we set to 8 or 16 in our examples. We can then generate a variety of images from the same CPPN by sampling this vector from a normal distribution $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.

iii. These coordinates are then fed through a neural network (Figure 3.1. Here we simply use a series of L linear transformations followed by a nonlinear activation function. For layer l, let $\mathbf{W}_l$ and $\mathbf{b}_l$ be the weight matrix and bias vector, respectively, and $f_l(\cdot)$ be the activation function. The linear transformation at each layer is:

$$\mathbf{h}_l = \mathbf{W}_l \mathbf{a}_{l-1} + \mathbf{b}_l$$

and the activation function at layer l is:

$$\mathbf{a}_l = f_l(\mathbf{h}_l).$$

In our implementation we use tanh. though common choices for f_l include the ReLU, or sigmoid function. A variety of architectures are explored in Figure 3.3, where the influence of the parametrization of the model and the image quality is visualized.

iv. The output layer produces a 3-channel vector representing RGB values of the color at $\mathbf{x}$. A 1-channel or 4-channel vector can be used to represent black and white or RGBA values, and higher dimensional outputs can be used to represent 3D objects [79], or hyper-networks, networks that 'draw' other networks [72–74]. A sigmoid is used on the final layer to bound the values between $[0, 1]$ in order to map them to RGB values.

v. We can parallelize the computation of an entire grid of coordinates since each coordinate's output is calculated independently. In our implementation in PyTorch [80] we achieve this by flattening our spatial dimensions into the batch dimension of our tensors.

vi. We can write this model in a differentiable way for optimization purposes. In Figure 3.4 a variety of architectures are trained on a set of target images using a mean-squared-error loss function. Architecture choice has a very strong impact on the trainability of these networks and also the resulting image quality[1].

3.1.2 Autocorrelation

Autocorrelations are the correlations a variable has with delayed (spatially or temporally) versions of itself. Autocorrelation *functions* (acf.) define the autocorrelation of a sequence or an image as a function of some distance parameter **d** (e.g. radial distance $d = r$ or angle $d = \theta$).

Let $\mathbf{I}(n, m)$ represent a 2D image of size $n \times m$. Taking $d = (\Delta x, \Delta y)$ where Δx and Δy represent the number of pixels separating two points, the acf. of this image $\rho(\Delta x, \Delta y)$ at a specific distance is computed by applying the expression:

$$\rho(\Delta x, \Delta y) = \sum_{\{n,m\}} I(n, m) \cdot I(m + \Delta x, m + \Delta y)$$

which sums over all possible pixels (n, m). In practice, this convolution operation can be accomplished much faster with Fast Fourier Transforms. By taking the Fourier transforms of the image, a multiplication with its flipped version will compute its autocorrelation.

The function can be parameterized as a function of radial distance $\mathbf{r} = \sqrt{(x^2 + y^2)}$ or angle $\theta = \arctan(y, x)$, shown in Figure 3.5 for images generated by different architecture CPPNs. Autocorrelations of images are used in the textile and materials industry for the analysis and quality control of material products [81–84] and are an informative quantitative tool for image classification.

Generally, the acf. reflects repeating statistical properties of a system and can therefore be used in a variety of applications where spectral quantities are important, such as frequency detection or detecting anisotropy to name a few. Intuitively, a system that generates a quickly decaying acf. has very little self-similarity, such as in a noisy, random or chaotic signal versus a system that has large and slowly a slowly decaying acf. which would have to be much more structured and change more smoothly. The decay rate of acf. in data or nature can often be fit with exponential functions which have characteristic timescales related to their decay rates. However, there are some instances where systems exhibit acfs. with no characteristic timescales which decay according to power laws such as in the 2D Ising model at criticality [85, 86]. In the context of CPPNs generating aesthetic imagery, we simply use the computed acfs. as features to correlate with the aesthetic experience of an image. Our hypothesis being that aesthetic images and experiences toe the line between complexity and order, a property that is often associated power-law statistics and criticality [87].

[1] We don't train CPPNs in our publication, however it is possible to do so. In Figure 3.4 it can be seen how different architectures can constrain the image properties uniquely.

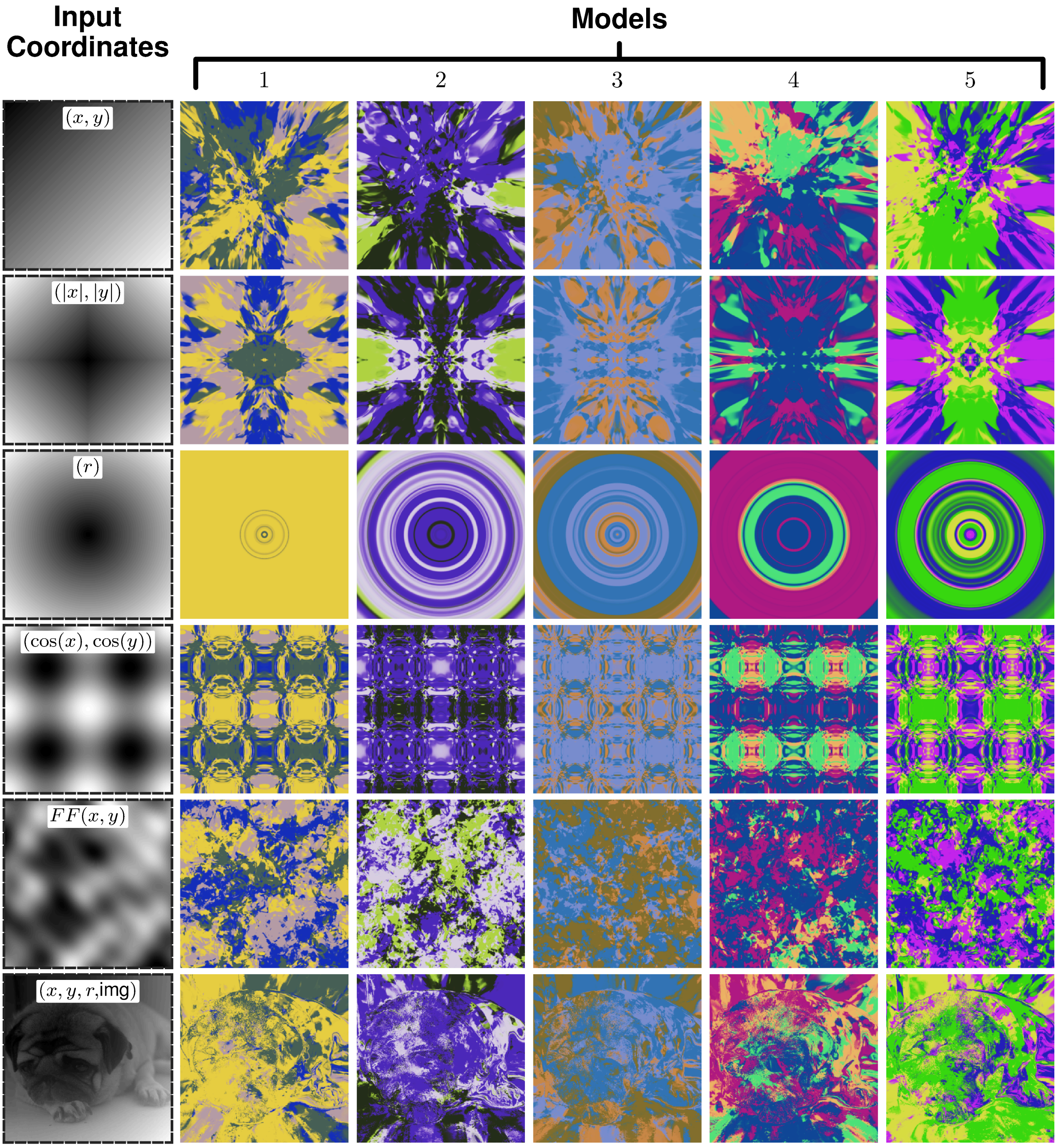


Figure 3.2: Image structure as a function of coordinate space. The images generated by a CPPN have structure, despite the fact that each pixel is processed independently, by the nature of the structured coordinate space fed as input. The images generated in each row by 5 instances of a single architecture are a function of the input coordinates shown in the left-most column. As we vary or augment the input coordinate vector with different coordinates, we can see how the generated image reflects the structures of these coordinate spaces. The essence of CPPNs is visualized here explicitly, that a coordinate axis with no interacting components and random matrix multiplications are readily capable of generating complex yet structured patterns.

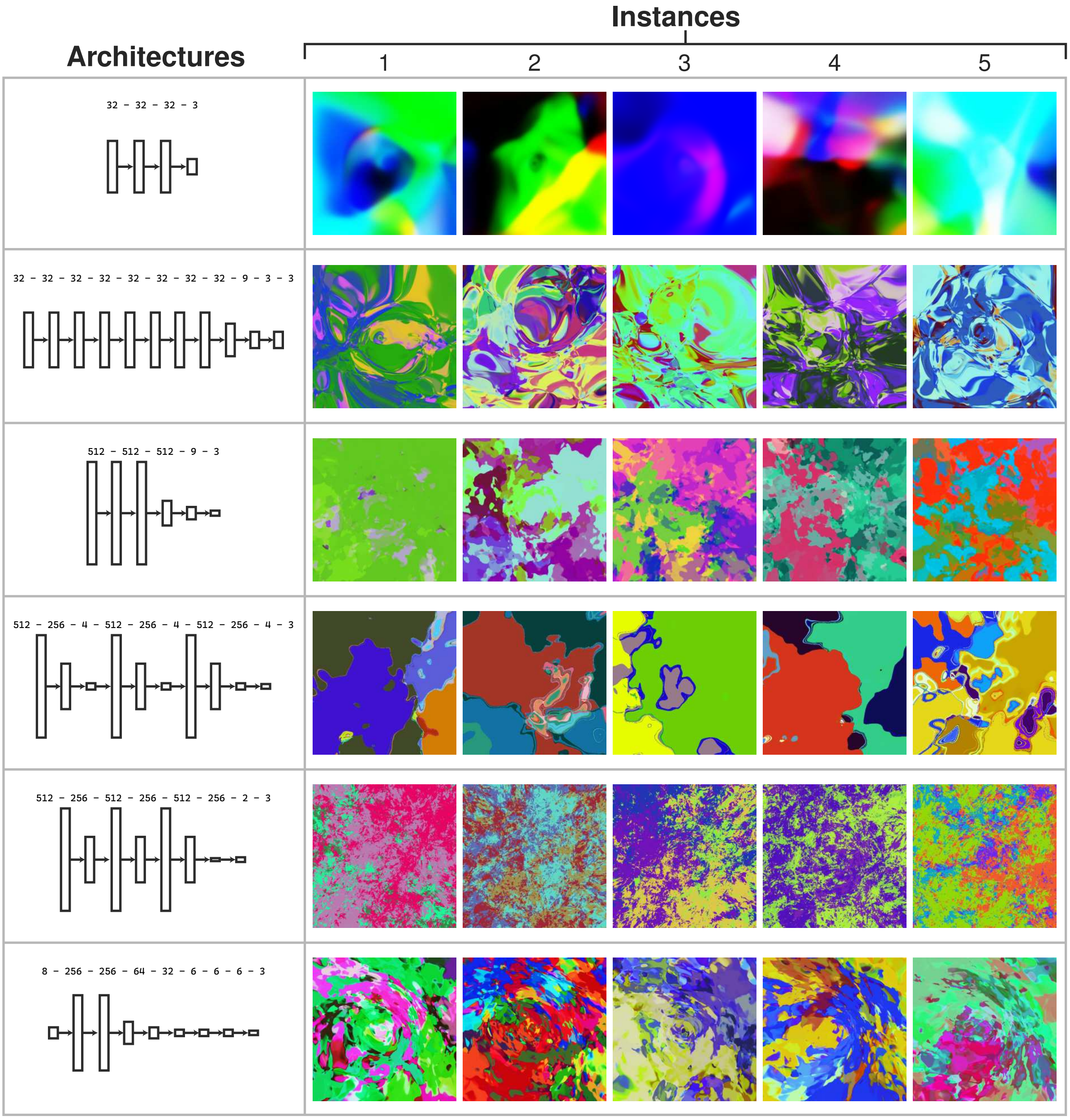


Figure 3.3: Image quality as a function of network architecture. The left-most column lists a number of selected architectures that are used to generate a set of images with 5 random instantiations of the model. The architectures are selected to demonstrate a variety of shapes and bottlenecks (or lack thereof) that result in diverse qualities of images. As we increase the number of neurons and layers, the texture frequency tends to increase. Bottlenecks act as high-pass filters and seem to be one of the key ways to induce correlations and to generate aesthetically pleasing images. Combining these motifs can form interesting structures in the resulting images.

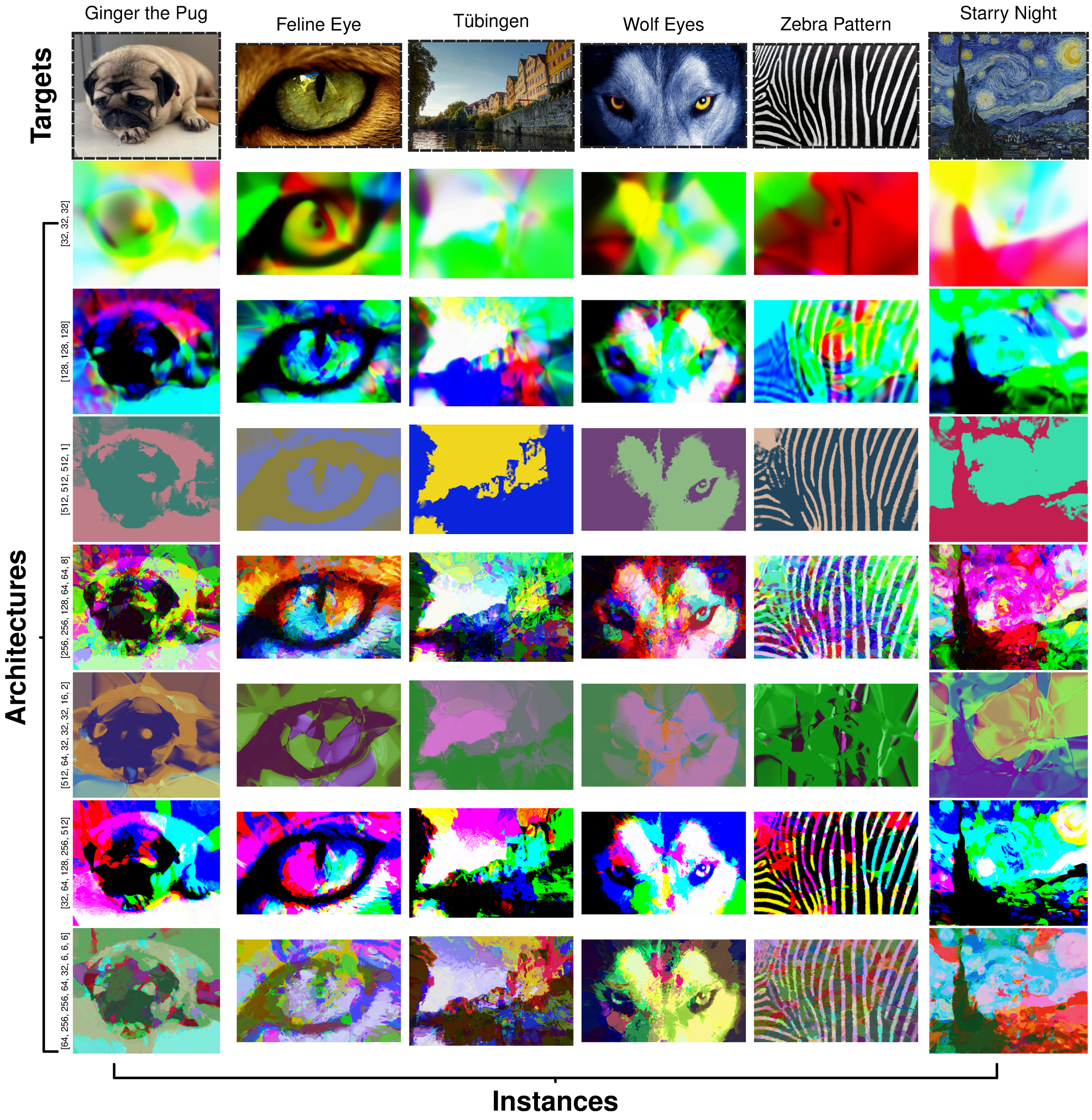


Figure 3.4: CPPNs trained on image reconstruction. CPPNs of different architectures are optimized with back-propagation and gradient descent to reconstruct target images (top row). A mean square error between the target and output image are used as a loss function. The images are averaged across their RGB dimensions when calculating the loss in order to allow the network to have some 'creative freedom' in colouring in the images.

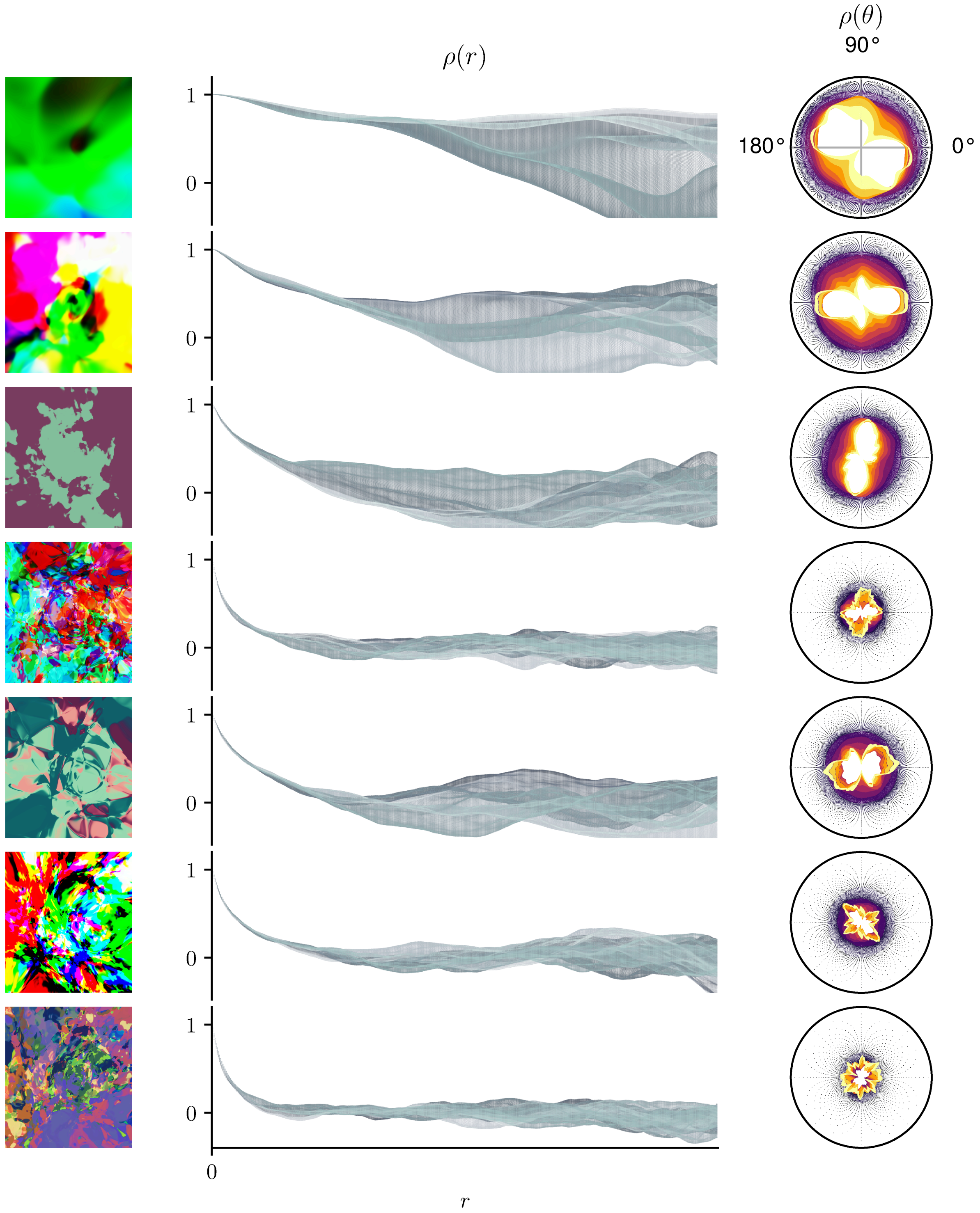


Figure 3.5: Auto-correlation functions of 2D images. The auto-correlation function $\rho(\cdot)$ of CPPN generated images (first column on the left) are computed and plotted as scatter plots as functions of radial distance (r) (second column) and angle (θ) (third column). Smoother, less noisy images generated by smaller or simpler architectures tend to have larger correlations that decay more slowly, shown in the second column. Furthermore, simpler architectures or those with strong bottlenecks tend to produce more anisotropic images with large structures as seen by the asymmetries in the polar correlation function in the third column.

3.2 The Ising Model, Phase Transitions, and Critical Points

> *At the time I wrote my doctor thesis Stern and Gerlach were working in the same institute on their famous experiment on space quantization. The ideas we had at that time were that atoms or molecules of magnets had magnetic dipoles and that these dipoles had a limited number of orientations* [88].
>
> Ernst Ising

3.2.1 Historical review

Before introducing the Ising model, it is helpful to first introduce the concepts of phase transitions and critical points. Broadly speaking, a **phase transition** describes the phenomena of a sudden transformation of a system. It is a relatively ubiquitous phenomenon, for example when water boils to steam, or freezes into ice, it undergoes a phase transitions from a liquid phase to a gaseous or solid phase. Though the molecular components of the bulk material are identical, H_2O molecules, the macroscopic system has reorganized itself in such dramatic fashion to have completely new bulk properties. These properties are then stable for a relatively broad range of parameters (for example liquid water can exist from 0°to 100°at atmospheric temperature, but will spontaneously freeze or boil outside that range).

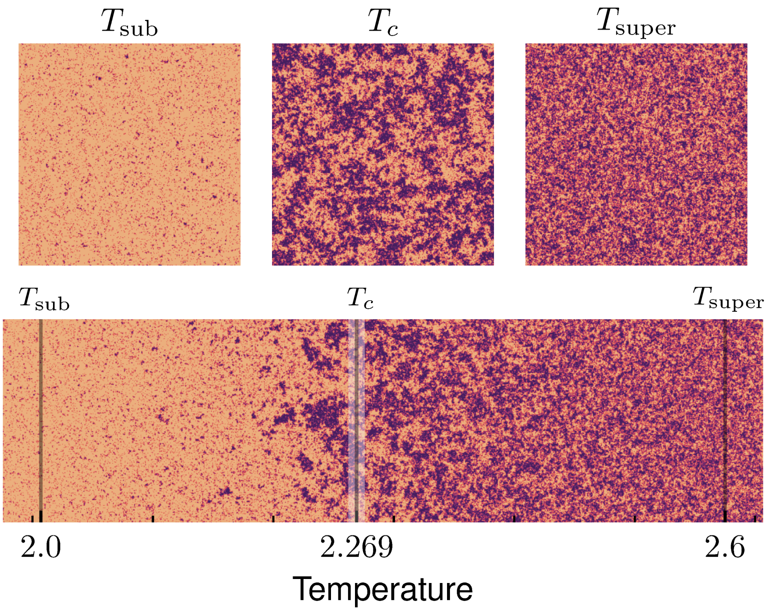


Figure 3.6: Spin configurations of the 2D Ising model at different temperatures. The spins $s_i \in \{-1, +1\}$ are shown in dark and light colours in equilibrium configurations. **Top row:** The Ising model is simulated on a 512×512 grid for $2{\times}10^4$ iterations using the Metropolis algorithm [89] for 3 temperatures below, at, and above the known critical temperature of $T_c \approx 2.269$. **Bottom row:** Similar to the top row but with size 1920 × 512 and a continuous range of temperatures along the x-axis in order to visualize how narrow the parameter range for critical behaviour is.

There are different types of phase transitions, and one common way to categorize them is according to the order in which the discontinuity of the system lies. If the order parameter of a model (e.g. the net magnetization M) has a discontinuity as a function of its control parameter (e.g. the temperature T), it is said to have a first-order phase transition. If the order parameter remains continuous under such changes and it is instead its derivative that is discontinuous, it is said to have a second-order phase transition and a **critical point**[2]. A spectrum of exotic behaviours occur near the critical points of second-order transitions, where thermodynamic variables such as the heat capacity or the susceptibility of a material diverge, resulting in properties such as diverging correlation lengths and critical slowing down. In recent decades, the anomalous behaviour of materials near their critical points is a point of heavy interest in explaining adaptive or computational systems. Predicting the conditions which give rise to phase transitions has historically been a difficult task, and it is Ernst Ising who begins to tackle this endeavor for his PhD thesis in order to explain the behaviour of permanent magnets.

Historically, the invention of the Ising model was to explain the strange phenomenon of ferromagnetism and to understand how bulk magnetic properties can spontaneously emerge (or vanish) in some metals. It was known at the time that heating a magnet past its Curie temperature would result in a phase transition from a ferromagnetic to a paramagnetic phase, where the material's permanent magnetic qualities would vanish. However, the microscopic mechanisms of this phenomenon were not understood, and therefore a model capable of demonstrating a phase transition was needed. The model emerged at a time when the quantum understanding of physics was only beginning to be articulated, where now fundamental concepts such as electron spin were still new and poorly understood[3].

Interestingly, Ernst Ising showed that no such phase transition exists in the solution to the 1D Ising model, and that there was no non-zero temperature where the 1D chain of spins could conspire to form an ordered state. However, even at the time many prominent physicists (including Wolfram Pauli) would discuss the possibility of a phase transition to emerge if these results were to be extended to higher spatial dimensions [52]. It would take a few more years until Lars Onsager would analytically solve the 2D Ising model, showing it *does* indeed have a phase transition and a critical point.

Initially criticized as too unphysical to model anything real [51, 52], the Ising model has since proven to be foundational to the study of collective phenomena. In describing systems near their critical points, it became clear that many disparate systems of varying microscopic details can exhibit identical bulk properties, a phenomenon called *universality*. Thus, the strange properties that arose around critical points and phase transitions could be extended to a much broader set of collective phenomenon than just the specific one studied (magnetism, in the case of the Ising model). Today, these ideas are being explored in even broader domains, where biological systems such as gene-expression networks, neural networks, or socio-economic networks are suspected to exhibit properties shared by critical phenomena.

3.2.2 Model

The Ising model is composed of a set of binary *nodes* or *spins*, $s_i \in \{-1, +1\}$ that are arranged into a lattice. Spins interact with their nearest neighbours, where the interaction strength is defined by the connectivity

[2] The 1D Ising model has a first-order phase transition when the temperature is $T = 0$, whereas the 2D Ising model has a second-order phase transition at a critical temperature $T_c \approx 2.2691$ (see Figure 3.6.

[3] The famous Stern-Gerlach experiments conducted in 1922 would eventually lead to the discovery of the quantum phenomena of **spins**, an intrinsic and quantized unit of angular momentum carried by elementary particles.

matrix J_{ij}. Furthermore, the spins can interact with an external field, h. All together, the energy of an Ising lattice is given by:

$$E(s) = -\sum_{\langle ij \rangle} J_{ij} s_i s_j - h \sum_i s_i, \tag{3.1}$$

where the sum over $\langle ij \rangle$ represents a sum over nearest neighbours.

- When $J_{ij} > 0$, the spins interact *ferromagnetically* and prefer to align themselves to their neighbours.
- When $J_{ij} < 0$, the spins interact *anti-ferromagnetically* and prefer to be anti-parallel to their neighbours.
- When $J_{ij} = 0$, the spins do not interact and are independent from each other.

The probability for any configuration to be observed at equilibrium is given by the Boltzmann distribution:

$$p(s) = \frac{e^{-\beta E(s)}}{Z}, \tag{3.2}$$

where $\beta = \frac{1}{k_B T}$ is the inverse of the temperature multiplied by the Boltzmann constant k_B, and the partition function[4]:

$$Z = \sum_s e^{-\beta E(s)} \tag{3.3}$$

is used as a normalizing factor.

Partition Function

With the partition function (Eqn. 3.3), the expectation values of the system's observables can be computed. Note that

$$\begin{aligned} \langle E \rangle &= \sum_s p(s) E(s) && (3.4)\\ &= \sum_s E(s) \frac{e^{-\beta E(s)}}{Z} && (3.5)\\ &= \sum_s \frac{-\frac{\partial}{\partial \beta} e^{-\beta E(s)}}{Z} && (3.6)\\ \langle E \rangle &= -\frac{\partial \ln Z}{\partial \beta}. && (3.7) \end{aligned}$$

And in general if the energy of the system can be written in the form of $E_s = E_s^0 + A_s \lambda$, where λ is a parameter we can vary (such as the external magnetic field (h) in Eqn. 3.1), then we can compute expectation values of any property with the relation:

$$\begin{aligned} \langle A \rangle &= \sum_s A_s p(s) && (3.8)\\ &= -\frac{1}{\beta} \frac{\partial \ln Z}{\partial \lambda}, && (3.9) \end{aligned}$$

and taking λ to go to zero in the end. For example, in the Ising model with an external magnetic field h, the expected magnetization of

[4] 4: The partition function describes the equilibrium behaviour of a thermodynamic ensemble, from which many other thermodynamic properties like entropy or free energy can be calculated.

the system with zero external field is given by:

$$M = -\frac{1}{\beta}\frac{\partial \ln Z}{\partial h}\bigg|_{h=0} \tag{3.10}$$

which is a first-order derivative of the system.

Computing the partition function quickly becomes analytically intractable for complex systems and computationally expensive for models with large numbers of nodes. Instead, numerical methods exist to approximate these equilibrium distributions iteratively. Using Monte-Carlo methods, a random spin site s_i is chosen and the probability of its flip computed as a function of the change in energy the flip would result in. The flip is accepted with probability:

$$p(\Delta E) = \frac{1}{1 + e^{\beta \Delta E}}, \tag{3.11}$$

when using Glauber dynamics, or

$$p(\Delta E) = \begin{cases} 1, & \Delta E \leq 0 \\ e^{-\beta \Delta E}, & \Delta E > 0 \end{cases} \tag{3.12}$$

when using the Metropolis-Hastings algorithm. By iterating this algorithm over all spin sites repeatedly, the equilibrium statistics of the model can be computed (Figure 3.6). When simulating the Ising model or generalization thereof, it is common practice to measure its thermodynamic properties (Figure 3.7) which are known to have interesting behaviours near the critical point.

In general, for any model that could potentially exhibit criticality, one typically defines an **order parameter** (a macroscopic quantity, usually normalized between 0 and 1, that represents some symmetry or structure of interest, e.g. magnetization or density) to measure as a function of a **control parameter** (some externally adjustable parameter that can control the degrees of freedom of the system, e.g. temperature or pressure). Choosing an appropriate order/control parameter is non-trivial, where the order parameter is ideally some value that is 0 in the disordered state and 1 in the ordered state, and a control parameter that is capable of changing the degrees of freedom in the system to induce a phase transition between the disordered/ordered regimes.

In the case of the Ising model, we take the magnetization of the system $M = \langle s_i \rangle$ to be our ordered parameter, which is directly related to first-order derivatives of the partition function (Eqn. 3.10).

We are interested particularly in 2nd order (continuous) phase transitions that exhibit the interesting critical properties (discussed in Section 2.1 – The Criticality Hypothesis and The Edge of Chaos) related to diverging correlation lengths. The **specific heat** (**C**) and **susceptibility** (χ) are two of the classical thermodynamic properties of a system that tend to diverge near criticality (and are second order derivatives of the free energy of the system). For the 2D Ising model, the susceptibility and specific heat are given by:

$$\begin{aligned}\chi &= \left(\frac{\partial M}{\partial H}\right)_T \\ &= \frac{\beta}{N}\left(\langle M^2\rangle - \langle M\rangle^2\right),\end{aligned} \tag{3.13}$$

and

$$\begin{aligned}C &= \left(\frac{\partial E}{\partial T}\right)_H \\ &= \frac{k_B}{\beta^2 N}\left(\langle E^2\rangle - \langle E\rangle^2\right),\end{aligned} \tag{3.14}$$

where the partial derivatives of the magnetization and energy with respect to the external field and temperature, respectively, are related to their fluctuations. In the case of the susceptibility, we have that the fluctuations in magnetization to be proportional to the spin-spin correlations $\langle (s_i - \langle s_i\rangle)(s_j - \langle s_j\rangle)\rangle$ of the system.

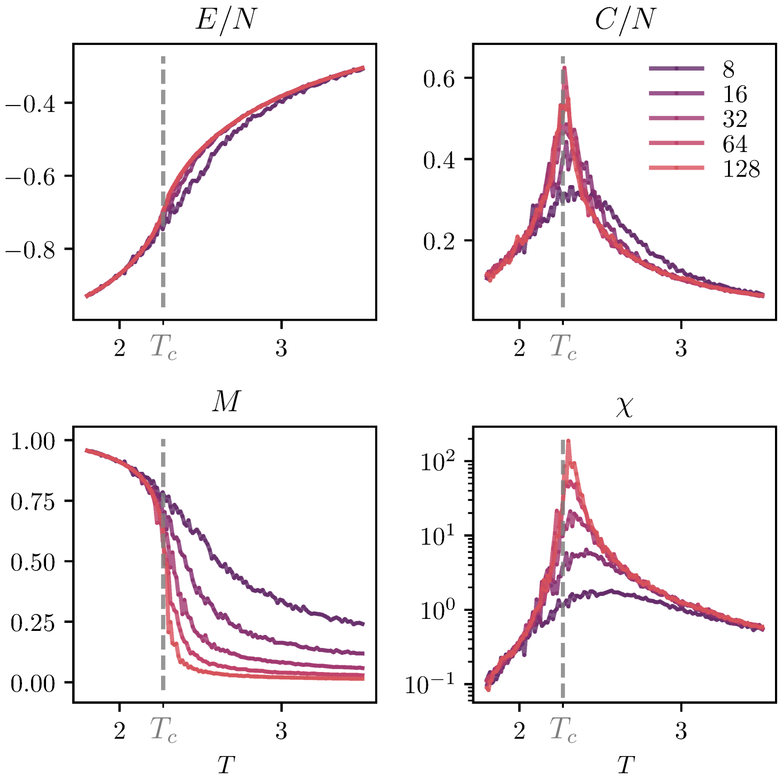


Figure 3.7: Statistics of the 2D Ising model on different sized grids and temperatures. Grids of length $L \in \{8, 16, 32, 64, 128\}$ are equilibriated for 10^3 iterations and then have their statistics: Energy (E), Magnetization (M), Heat capacity (C), and Susceptibility (χ) measured over 4×10^4 iterations every 5 steps. Statistics are averaged over 5 independent runs. The finite-size effects of the model can be observed as the peaks of the heat capacity and susceptibility curves shift towards the known critical temperature T_c when $L \longrightarrow \infty$.

Intuitively, the susceptibility is a measure of the sensitivity of the system from perturbations[5], and the specific heat is a measure of the energy cost to increase the temperature of the system. When approaching the critical point, these values diverge towards infinity, where these divergences give the characteristic properties of critical systems. As the system approaches the critical point, the diverging susceptibility of the system indicates that the correlation length diverges as well, meaning that small fluctuations can propagate (close to) infinitely near criticality. It is these diverging quantities that ultimately give rise to the exotic properties of critical systems but are also why disparate systems can be universal as their

[5] In the case of the Ising model, we care about how sensitive the magnetization (M) is with respect to a coupled external magnetic field (H). In general, however, these variables can be different for different systems.

defining features are related to the properties of their divergences rather than their microscopic details!

When utilizing maximum entropy methods to fit data, one typically constrains the model to the mean activity and two-point correlations of the dataset, $(\langle s_i \rangle, \langle s_i s_j \rangle)$, respectively. A model constrained on these properties will have the form:

$$P(\{s_i\}) = \frac{1}{Z} \exp\left[-E(\{s_i\})\right] \tag{3.15}$$

$$E(\{s_i\}) = -\sum_{i,j=1}^{N} J_{ij} s_i s_j - \sum_{i}^{N} h_i s_i \tag{3.16}$$

where h_i corresponds to the mean activity of a node $\langle s_i \rangle$ and J_{ij} corresponds to the two-point correlation $\langle s_i s_j \rangle$. It should be noted that Equation 3.16 is identical in form to Equation 3.1, the energy of the Ising model, though we no longer have a temperature (T) parameter when we are fitting. We can, however, reintroduce this parameter in order to make measurements of the model in the neighbourhood of the fitted parameters in order to make a plot similar to Figure 3.7. Doing so allows us to analyze the fitted model thermodynamically (see for example Figure 3.8 and [17]).

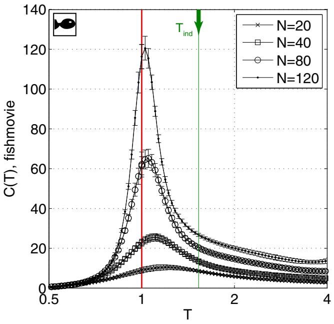


Figure 3.8: Heat capacity $C(T)$ in a maximum entropy model of neurons responding to naturalistic stimuli for subnetworks of size $N = 20, 40, 80,$ and 120 neurons. The peak near $T = 1$ is indicative that the fitted maximum entropy model falls onto a critical point. Figure taken from [17].

3.3 Neural Cellular Automata (NCA)

> *In pure mathematics the really powerful methods are only effective when one already has some intuitive connection with the subject, when one already has, before a proof has been carried out, some intuitive insight, some expectation which, in a majority of cases, proves to be right. In this case one is already ahead of the game and suspects the direction in which the result lies [90].*
>
> John Von Neumann

Neural Cellular Automata (NCA) are an extension/generalization of **Cellular Automata** (CA) that utilize neural networks as the operators in their rule set instead of (or in addition to) the logic operations that are utilized in classical CA.

Cellular automata are mathematical models of computation, usually composed of a 1D or 2D grid of cells of binary[6] values, each of which only interacts locally and updates its own state as a function of its neighbours (Figure 3.9). These models are typically very simple, being describable in just a few sentences (see below for the rules of Conway's Game of Life), and in the case of NCA, have a very low number of parameters[7] due to the fact that all parameters are shared across all cells in a grid[8]. Nowadays, these models are used to simulate a variety of complex systems in a range of fields, from computer science and physics[9] to biology and urban development [92–96]. Cellular automata were originally devised as a model of self-replication by Stanslaw Ulam and John von Neumann [90] who were interested in simulating self-replication and universal computation in models of simple, locally interacting cells. To this day, scientists interested in the emergence of complexity, computation or self-organization are still utilizing CA models to understand how simple systems can form complex structures, pattern, and ultimately, *life*.

3.3.1 Conway's Game of Life (GoL)

CA were popularized in the 70s after the discovery of "Conway's Game of Life" by John H. Conway [98]. This 2D CA had a rule set that attempted to achieve a balance between perpetual growth and death (which are common states to converge upon when exploring randomly generated CA rule sets). Conway discovered that by balancing these tendencies, that this automaton would give rise to *interesting*[10]dynamics and complex behaviour that would prove to be Turing complete. The rule set for Conway's Game of Life are listed below (along with their life-like metaphor):

- **"Death by Underpopulation":** Active cells with < 2 active neighbours become inactive in the next step.
- **"Life by Balance":** Active cells with 2-3 active neighbours remain active in the next step.
- **"Death by Overpopulation":** Active cells with > 3 active neighbours become inactive in the next step.
- **"Life by Reproduction":** Inactive cells with exactly 3 active neighbours become active in the next step.

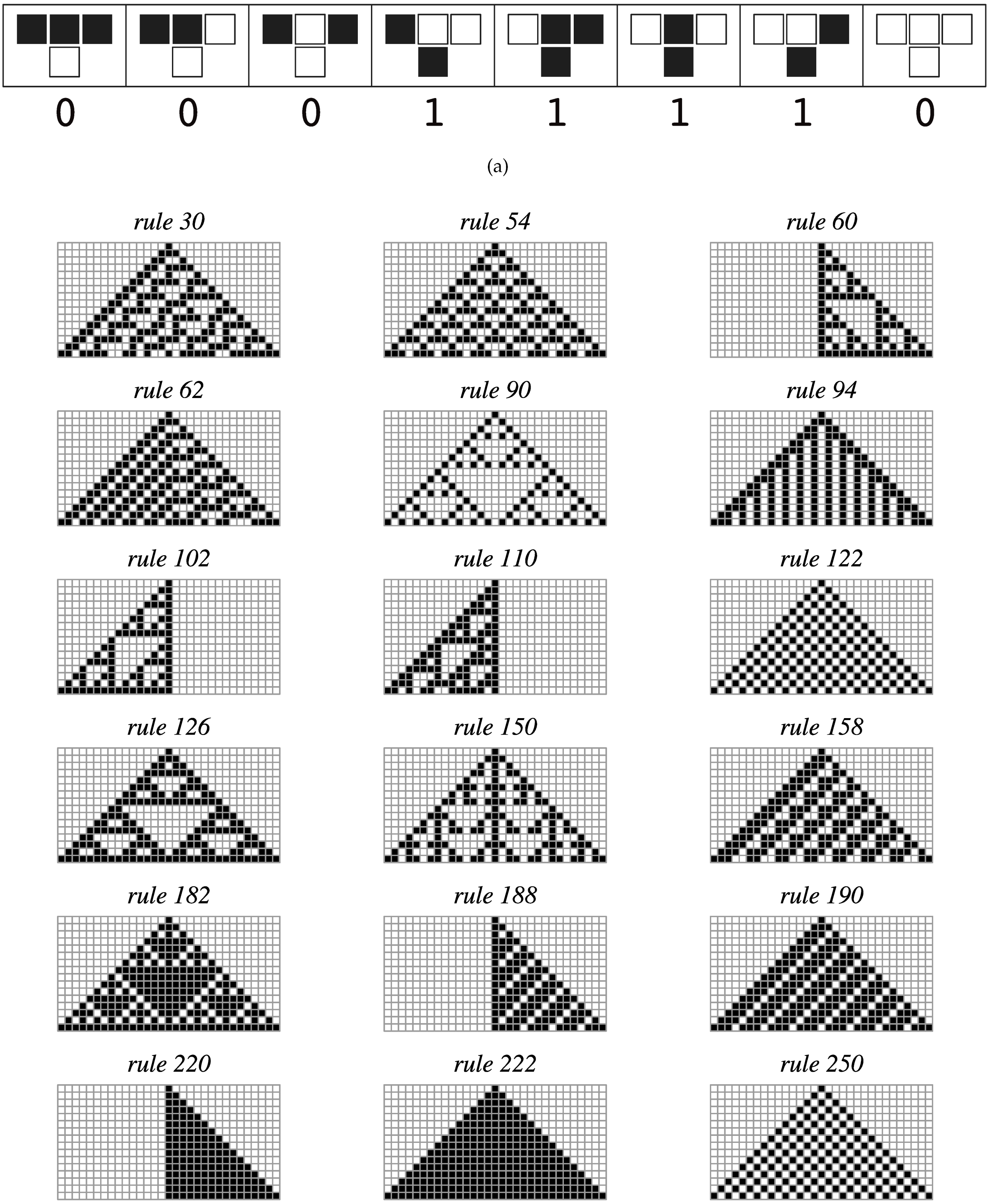
[6] Binary is the simplest case. There also exist n-state CA, similar to how the Potts model [91] generalizes Ising-like models to more than 2 states.

[7] When compared with deep networks, NCA share most parameters and generate complexity through recurrent feedback. Deep networks on the other hand often specify parameters for each component in the computational chain.

[8] Analogous to the uniformity of the laws/parameters of physics in space.

[9] The 2D Ising model introduced in the previous section can also be considered a (stochastic) CA.

[10] The word 'interesting' is quite subjectively loaded, but in the dynamical systems sense, it is a system which does not converge trivially to a static (or equivalently boring, completely random) state. Here, we are inspired by the far-from-equilibrium properties of living systems.



(a)

(b)

Figure 3.9: a): Example rule visualized graphically for an elementary CA. The update rule for the center cell is a function of its own state and the state of its nearest neighbours in 1D. **b):** Temporal evolution of a number of different 1D elementary CA rules. The horizontal axis represents space and the vertical axis represents time. Note how some rules converge to fixed states, some display fractal patterns, and some oscillate. Figures taken from [97].

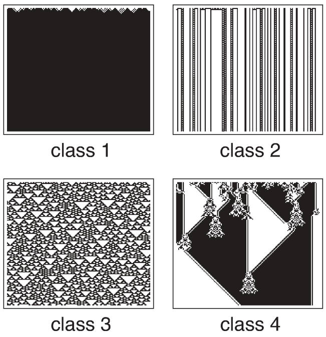


Figure 3.10: Image modified from [99]. Four examples of CA rules from different behavioural categories.

In the 1980s, Stephen Wolfram's work on cellular automata systematically explored a class of CA known as elementary cellular automata, enumerating all possible rule sets for binary 1D CA and classifying them according to their behavioural tendencies. Wolfram identified 4 main behavioural CA categories:

- Class I: rapidly converges to a uniform state.
- Class II: rapidly converges to a repetitive or stable state.
- Class III: appears to remain in a random state.
- Class IV: forms areas of repetitive or stable states, but also forms structures that interact with each other in complicated ways.

Often, practitioners are mostly interested in class IV CA for their ability to exhibit complex dynamics which GoL falls under. GoL enthusiasts have since developed a vast catalogue of objects [100] that exhibit interesting and increasingly complex behaviours: initial configurations which are stable and do not change (still life), patterns which repeat with some period (oscillators), patterns that can move across the grid (spaceships), or even simulate a computer running the Game of Life, or a calculator, or a clock (universal computers) [101–104].

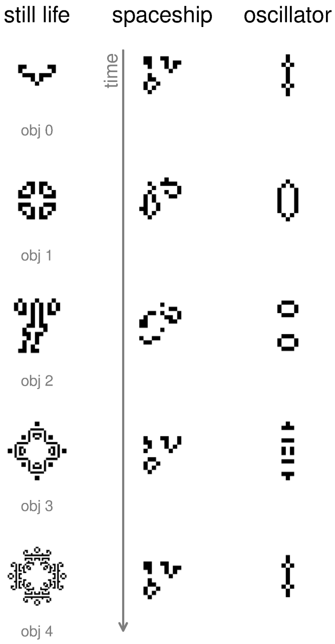


Figure 3.11: Examples of still-life, a spaceship (loafer, period 7), and an oscillator (pentadecathlon, period 15) pattern taken from [100]. These patterns were found to exist across multiple rules.

However, while there are still frequent discoveries being catalogued for GoL and other elementary cellular automata, a new branch of *neural* cellular automata has emerged, integrating the methods and success of modern machine learning with these simple models.

3.3.2 Convolutional Neural Networks (CNNs) and Cellular Automata

Neural cellular automata incorporate the techniques utilized in machine learning to compute and optimize CA rule sets for a desired behaviour. The basic idea is simple, to replace the rule set with a neural network that computes the new state of the system, where this neural network is applied to every cell in the grid. A convenient way to compute these operations efficiently with modern hardware is to use convolutions. In Figure 3.12, a convolutional implementation of Conway's Game of Life is shown [105].

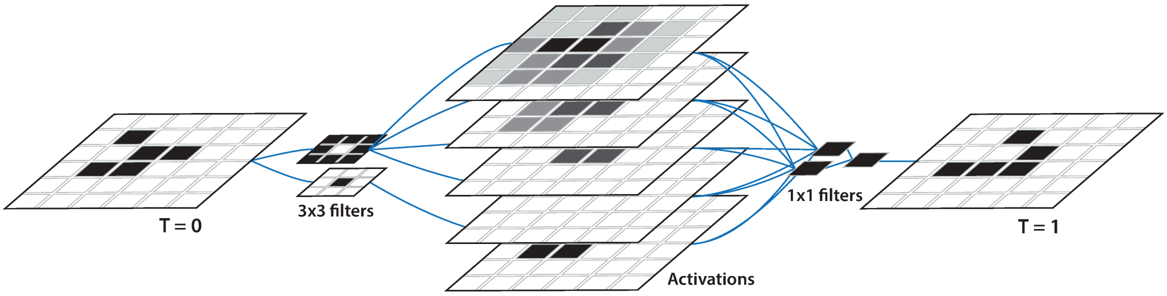


Figure 3.12: The game of life as a convolutional neural network. The update rule is replaced with sets of kernels and biases which can update each cell in parallel, taking advantage of GPU hardware. Figure taken from [105].

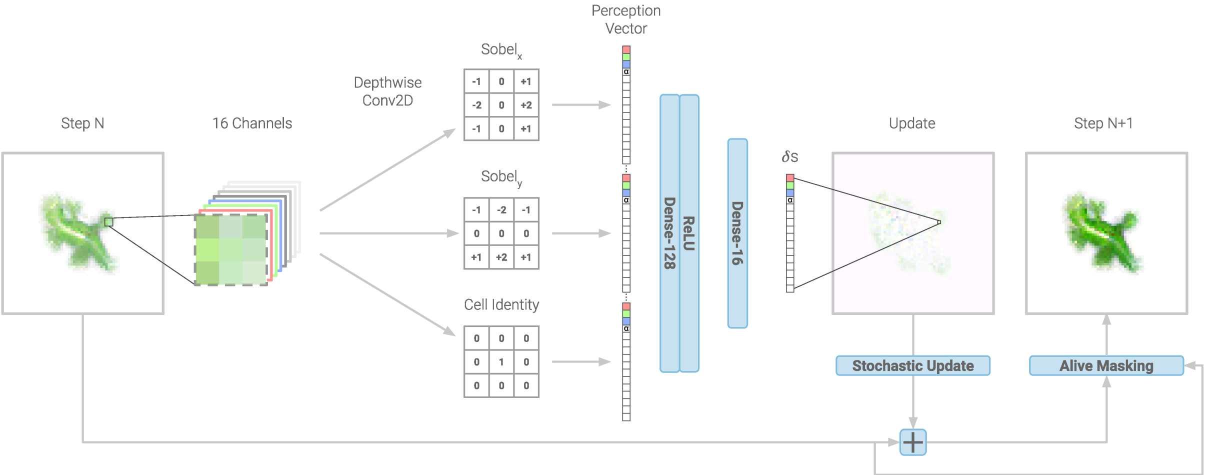


Figure 3.13: An example of an NCA architecture taken from [106]. The state tensor is *perceived* by a set of fixed kernels which are then passed through a series of neural network layers in order to produce an update value. This type of architecture is differentiable and can therefore be trained with standard optimizers such as stochastic gradient descent.

Writing the state of a CA grid at time t as $\mathbf{s}^{(t)}$, the new state $\mathbf{s}_{i,j}^{(t+1)}$ of the cell at position (i, j) at time $t+1$ is calculated as follows:

$$\mathbf{s}_{i,j}^{(t+1)} = \sigma\left(\mathbf{K} \otimes \mathbf{s}^{(t)}\right)_{i,j} \tag{3.17}$$

$$= \sigma\left(\sum_{m=-1}^{1}\sum_{n=-1}^{1} \mathbf{K}_{m,n} \cdot \mathbf{s}_{i+m,j+n}^{(t)}\right) \tag{3.18}$$

where $\mathbf{K}_{m,n}$ are the weights of the convolution kernel at the relative position (m, n) and $\otimes$ denotes the convolution operation. The function $\sigma(\cdot)$ is typically a nonlinear activation function such as the ReLU function (Rectified Linear Unit), a Sigmoid, or Tanh. Written this way, we can efficiently update the state of each cell in parallel using highly optimized linear algebra libraries. Furthermore, these convolutions can be chained together, for example to form deep neural networks:

$$\mathbf{s}^{(t+1)} = \sigma^{(L)}\left(\mathbf{K}^{(L)} \otimes \left(\cdots \sigma^{(2)}\left(\mathbf{K}^{(2)} \otimes \left(\sigma^{(1)}\left(\mathbf{K}^{(1)} \otimes \mathbf{s}^{(t)}\right)\right)\right)\cdots\right)\right),$$

where $\mathbf{K}^{(L)}$ are the different kernel weights used by each layer L. Furthermore, by allowing the state grid $\mathbf{s}$ to be a tensor, we can also allow for each cell to have C channel dimensions which can be used to extend the number of properties that define a cell.

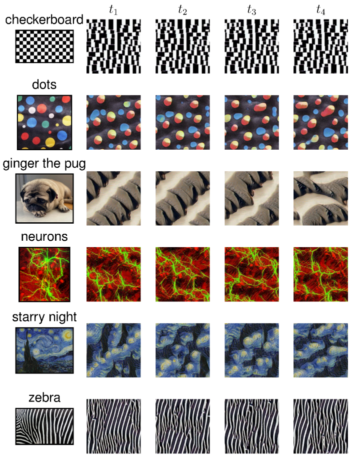


Figure 3.14: Neural cellular automata trained as self-organizing textures of images [107]. Implemented from `https://distill.pub/selforg/2021/textures/`

The extension of CA to NCA (Figure 3.13) has reinvigorated the research behind the generation and control of complex and emergent systems by utilizing deep learning techniques to optimize these distributed systems. While these models are still not competing versus state-of-the-art models in classic machine learning use-cases, they have demonstrated new capabilities in optimizing local[11] collective systems for global/macroscopic goals (see for example [107, 108], and an example implementation in Figure 3.14).

In our publication Section 4.3 – Locally adaptive cellular automata for goal-oriented self-organization we utilize this method of modeling CA as convolutions and loosen the restriction that every cell obeys the same

[11] State-of-the-art deep networks are usually not local in the sense that information can propagate arbitrarily far, whereas in CA, information must be propagated by local action.

update rule by introducing *heterogeneous* update rules. While this method has heavier memory requirements, we generalize the implementation of NCA to allow for adaptation and self-organization which is especially useful in cases with local and heterogeneous driving forces. We utilize these methods to simulate a 1000×1000 grid of self-organized to criticality Ising model as well as plastic spiking neural networks that run on a laptop GPU (NVIDIA GeForce RTX 3080).

3.4 Plasticity in Neural Networks

There are approximately 80 billion ($\sim 10^{10}$) neurons in the average human brain and as many as 100-1000 trillion ($\sim 10^{14}-10^{15}$) synapses that connect neurons across their dendritic and axonal trees. These trees receive and send signals, respectively, to other neurons in their vicinity. Individual neurons propagate signals across their axons which are typically sharp impulses or spikes, and can induce similar spikes in other neurons that receive sufficient input from their dendrites. Models of neural networks typically fall into one of two categories, spiking models or rate models, where the former category details the behaviour of individual spikes in neurons and the latter details the mean rate of firing of a neuron averaged either over a temporal or spatial window. In order to facilitate learning and the emergence of structured and behaviourally complex neural networks, some mechanisms for adaptation and change are necessary to tune these systems towards a functional or optimal state.

The brain is constantly undergoing structural changes throughout our lifetime while we develop and learn, where numerous mechanisms for adaptation across multiple timescales in the human brain have been studied extensively [109, 110]. Some examples include: synaptic plasticity which acts on the synaptic strengths between neurons w_{ij}[12], threshold adaptation which controls the required amount of input signals necessary to facilitate a spike, or neural energetic models that explicitly account for the resource consumption and replenishment of neurons [115, 116], to name a few. Furthermore, in light of the critical brain hypothesis [18, 24], a growing body of work has shown dynamical synapses or judicious combinations of plasticity rules [117] (for example a Hebbian rule combined with an appropriate normalization) can give rise to self-organized criticality [115, 118, 119] (or self-organized quasi-criticality [120, 121]) as measured by the avalanche statistics of these systems.

Strangely, these plasticity mechanisms do not perform competitively when compared to current state-of-the-art machine learning models (e.g. Transformers, MLPs, LSTMs) that do not employ plasticity mechanisms. Instead, end-to-end differentiation utilizing back-propagation algorithms in conjunction with gradient-descent and supervised learning[13]are the methods of choice in practical settings. In other words, the synaptic changes necessary to facilitate learning are all done offline in machine learning, require complete knowledge of the computational graph of the network, differentiation through these chain of operations and need explicit objective functions/losses.

Generalized Hebbian, Oja, and STDP Rules

The most common form of synaptic plasticity is **Hebbian learning**, first proposed by Donald Hebb in 1949 [126], commonly phrased as "neurons wire together if they fire together" [127]. It can be formalized as:

$$\Delta w_{ij} = r_i \cdot r_j, \tag{3.19}$$

where r_i and r_j are (functions of) the activities of neurons i and j. This form of the rule is unstable and requires some external

[12] There are numerous ways synaptic plasticity can be modulated, for example via short-term [111] or long-term-potentiation or depression [112], to achieve homeostaticity, or due to spike-timing-dependent-plasticity [112–114].

[13] Unsupervised or intrinsic learning algorithms also exist to understand models of self-organized learning [122–125].

normalization methods to keep the weights bounded. However, there are other variations that take this into account more explicitly, such as **Oja's rule**:

$$\Delta w_{ij} = r_i \cdot r_j - r_j^2 \cdot w_{ij}, \tag{3.20}$$

where the second term ensures the synaptic strengths don't run-away.

While these forms of rules connect neurons that are correlated, they are agnostic to causal relationships which is where **Spike-Timing-Dependent Plasticity (STDP)** comes in:

$$\Delta w_{ij} = \begin{cases} A_+ \exp\left(\frac{t_j - t_i}{\tau_+}\right) & t_j < t_i \\ -A_- \exp\left(\frac{t_j - t_i}{\tau_-}\right) & t_j \geq t_i \end{cases} \tag{3.21}$$

where $t_{i,j}$ are the latest spike-times of neurons i, j, $\tau_\pm$ are the time-constants for the potentiation/depression, and $A_\pm$ are the amplitudes of the change which can be functions of the current weights $A_\pm = A_\pm(w_{ij})$.

While these generalized forms already cover a wide range of possible plasticity rules, many variants still exist, including *reward-modulated* plasticity, which we discuss in Section 4.4.

Thus, there exists a major discrepancy between how learning is accomplished *in vivo* versus *in silico*. Although some of these discrepancies may stem from our hardware choices for computation, the fundamental issue remains that we are not yet able to replicate the sophisticated levels of learning observed in the brain when using biologically plausible models. To address this, we utilize plasticity rules in two of the projects presented in this dissertation: fast implementations of plastic spiking and rate neural cellular automata with heterogeneous plasticity and threshold adaptation (see Section 4.3 – Locally adaptive cellular automata for goal-oriented self-organization), and embodied and evolved plastic neural agents that must forage to survive (see Section 4.4 – Environmental variability and network structure determine the optimal plasticity mechanisms in embodied agents). The objectives of these studies are, respectively, to develop fast algorithms for large-scale implementation of plastic neural networks and to investigate the performance, convergence, and sensitivity of plasticity rules in applied settings such as an embodied foraging game.

Publications

Publications 4

4.1 Assessing Aesthetics of Generated Abstract Images Using Correlation Structure

Khajehabdollahi, S., Levina, A., Martius, G. (2019). "Assessing Aesthetics of Generated Abstract Images Using Correlation Structure," 2019 IEEE Symposium Series on Computational Intelligence (SSCI), Xiamen, China, 2019, pp. 306-313 doi:10.1109/SSCI44817.2019.9002779. arXiv:2105.08635.

4.1.1 Summary

Are there universal statistical properties shared across aesthetic visual stimuli? The emergence of one's sense of *beauty* and *aesthetics* is complex and poorly understood, but it seems certain that it is a subjective and dynamically defined sense. This sense seems to be tightly contingent on both our personal and cultural history, and yet there also seem to exist universal, statistical patterns across cultures that are considered beautiful. Some common themes that arise in the discussion of the beauty in art are the *tension*, *contrast*, and the interplay between *expectation* and *surprise*, or between *patterns* and *novelty*. These concepts describe the complexity and inter-relationships of an object, some of which map to quantifiable statistics. In this project, we explore the specific question: can the beauty of an image be related to its correlation function? We generate sets of random, abstract images using Compositional Pattern Producing Networks (CPPNs), across a variety of architectures to generate a diverse set of images. We then calculate the auto-correlation functions of these images and measure their aesthetic score by surveying 45 volunteers. Finally, we relate the auto-correlation functions of the images to their aesthetic score, finding that more slowly decaying functions tend to be selected as aesthetic more often.

This project initially grew out of the serendipitous discovery that random images generated with CPPNs were often perceived as beautiful abstract images. The networks behave similarly to a shader[1] a type of software used to quickly and procedurally generate textures for computer graphics on GPUs, often using very simple/highly optimized code running in parallel. Naturally, the complexity of the generated images can be controlled in different ways by manipulating the architecture of the network (*e.g.* introducing bottlenecks, varying the number of layers or number of neurons). Now and then the generated images come out "ugly" or "boring", while others come out "beautiful" to the eyes of many practitioners [76, 129–131]. We parameterize the architecture of the CPPN and sample it such that we span the space of images that are subjectively deemed "ugly" or "boring" to those that are deemed "beautiful" or "interesting". We then design an experiment that generates a large variety of images

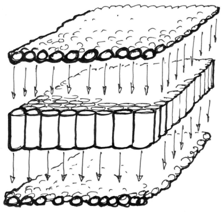


Figure 4.1: Figure taken from The Book of Shaders [128] as a caricature of how shaders generate images in parallel as a function of their pixel location. CPPNs operate in a similar fashion, where each pixel is associated with a coordinate vector and has its output values computed as a function of this vector.

[1] Shaders are the bread and butter of the computer graphics world, especially used in video games that require real-time rendering.

which are judged by human participants on their aesthetics and collect image statistics for aesthetic and non-aesthetic images.

A recurring phenomena in the statistics of nature and natural images is the occurrence of power-law distributions, often in their Fourier spectra. We therefore test if the difference between aesthetic images versus non-aesthetic images can be predicted by their correlation functions [2] and the similarity of these functions to those arising from natural imagery.

4.1.2 Results

Influence of network architecture on image statistics: We demonstrate how our architecture parameterization (a set of parameters controlling the number of layers, neurons, and shape of the neural network underlying the CPPN) generates models that output a broad distribution of image statistics. This is measured by their radially-averaged, grayscale, two-point correlation function. We find that a variety of correlation functions of different characteristic decay rates can be generated by manipulating the location and size of bottlenecks. This finding allows us a method to generate a variety of images to test for their aesthetic score.

Features of aesthetic images: Notably, most colour images tagged as aesthetic were generated by the smallest architectures (fewest neurons and layers). The grayscale images however had a slightly different distribution of aesthetic architectures where the mean preference skewed towards more neurons and therefore more fine-grained textures. Overall, the images deemed aesthetic had on average more slowly decaying correlation functions than the set of all CPPN generated images.

Comparison between generated images and natural images: Natural images themselves have a broad distribution of statistical properties that are a function of the scene and method of capture. We find that our set of CPPN generated images were able to produce a similar variety of distributions when using different architectures. For example, small, shallow networks had properties closer to that of the *LandWater* category of natural images while larger, deeper networks were more similar to images of flower or foliage. Interestingly, many participants in our survey reported seeing faces or animals in the random images (partially a by-product of the use of symmetric coordinate functions as input for our image generator, similar to the symmetric images of ink-blot tests.).

4.1.3 Discussion

This research explores the relationship between image statistics and the human perception of aesthetics and highlights the capabilities of CPPNs to generate a wide array of abstract images with various aesthetic and statistical qualities. While limited by a small sample size of participants, our study suggests that images voted as aesthetic have, on average, auto-correlations that decay more slowly (i.e. smoother, more structured imagery) than the full corpus of random images presented in the survey.

The ability for these randomly generated images to elicit sensations of beauty is slightly[3] unexpected for an untrained, randomly initialized model. How is it that a series of random matrix multiplication operations

[2] Autocorrelations are related to Fourier spectra and power spectral densities by the Wiener–Khinchin theorem.

[3] I write "slightly" because in many respects, random number generators are a key component in shaders and computer graphics which are responsible for producing beautiful and life-like visual effects. So perhaps ot shouldn't be too unexpected given how ubiquitously modern computer graphics utilize randomness.

can give rise to such patterns that the eye finds appealing? The perception of beauty out of randomness, or random processes, is scientifically inspiring because it implies that there might be mathematical and predictable properties associated to the phenomena we tend to find beautiful. In other words, it may be that something as personal and abstract as beauty could have a mathematical structure.

4.1.4 Author Contributions

- **Sina Khajehabdollahi** was responsible for code required to generate models, generate images, compute auto-correlations and all the analyses included in the paper.
- **All authors** contributed to the writing and reviewing of the paper, figure creation and editing, and performing surveys for aesthetic scores on images.

4.2 When to be critical? Performance and evolvability in different regimes of neural Ising agents

Khajehabdollahi, S., Prosi, J., Giannakakis, E., Martius, G., Levina, A. (2022). When to be critical? Performance and evolvability in different regimes of neural Ising agents. Artificial Life, 28(4), 458-478. doi:10.1162/artl_-a_00383. arXiv:2303.16195.

4.2.1 Summary

Is operating near a critical state advantageous for evolutionary systems? The hypothesis that it *is* has been supported by observations in various biological systems, like gene regulatory networks, neural networks, cell behaviour, and agent swarms. The accumulating evidence suggests a strong connection between criticality and optimal performance, adaptibility, and evolvability.

By grounding the thermodynamic and mathematical concepts of criticality into a functional, artificially-embodied agent learning to navigate its environment, this project allows us to test questions about the utility of criticality to evolutionary fitness. Is there an evolutionary attractor towards these types of systems? Is there some privilege to being near a critical dynamical state? Can we measure the effect the distance to criticality has on the performance of embodied agents? We devise experiments to answer these questions and test if operating near criticality is advantageous for evolutionary systems.

4.2.2 Results

Extensive numerical experiments with evolving Ising agents are run to test our hypothesis. The evolving agents are embodied in a 2D simulation where they must navigate a world to find food shared by other agents. The agents can move around by sensing their nearby world and controlling their rotation and velocity motors. Each agent has a small Ising neural network that controls their behaviour, and populations of agents are instantiated across a variety of dynamical states (with the control of an inverse-temperature parameter) and co-evolve together.

Evolution of the dynamical regime: All populations discovering solutions evolved to be subcritical. Interestingly, they did not converge to a critical state. However, *initially* critical populations found solutions much faster, sometimes by orders of magnitude in training time. Populations initialized more subcritical than the converging distance to criticality often failed to find solutions, indicating that being subcritical is a more difficult initialization than critical or even supercritical Ising agents, despite the best solutions converging to subcriticality eventually.

Generalizability and Robustness: Initially critical agents maintained their fitness level better under environmental changes and parameter perturbations when compared to initially sub-critical (but comparably) fit agents. We perturbed the weights of the Ising neural networks to test how sensitive/robust they were to damage. We also swapped agents trained in one environment into a more open-ended one. In both cases, initially critical agents sustained higher fitness when damaged and or when placed in new environments.

Task complexity correlates with distance to criticality: We introduce a new rule that agents must slow down to a threshold velocity in order to consume a food particle, forcing the agents to perform more deliberate sequences of behaviours in order to forage and attain energy and fitness. Populations training on this harder foraging task evolved to be closer to criticality on average than populations training on the simpler task.

4.2.3 Discussion

The study reinforces the significance of criticality in evolutionary systems. Despite optimal behavior in simple tasks being achieved in a subcritical regime, starting near criticality is crucial for efficiently finding solutions for new tasks of unknown complexity. Our resilience analysis, which showed that initially critical agents were more adaptable and robust under change, further supports this. The study also highlights the importance of considering task complexity in the evolutionary process, as the optimal state of criticality varies accordingly.

4.2.4 Author Contributions

- **Sina Khajehabdollahi** made significant contributions to the code base of this project, including its first iteration [132], as well as running experiments and data analysis.
- **Jan Prosi** has made significant contributions to the code base, expanding the variety of experiments and their analysis, including making significant additions in his follow up publication [133].
- **All authors:** contributed to the writing, discussion and editing process of this publication.

4.3 Locally adaptive cellular automata for goal-oriented self-organization

Khajehabdollahi, S., Giannakakis, E., Buendía, V., Martius, G., Levina, A. (2024). Locally adaptive cellular automata for goal-oriented self-organization. ALIFE 2023: Ghost in the Machine: Proceedings of the 2023 Artificial Life Conference. MIT Press, 2023. doi:10.1162/isal_a_00663. arXiv:2306.07067.

4.3.1 Summary

How can we make microscopic interactions give rise to desired collective behaviour? An intriguing aspect of collective systems is that their collective behavior often differs significantly from the properties of their individual components. The laws of physics, chemistry, biology, psychology, economics, and politics each operate differently, yet are interconnected. Understanding how microscopic rules at one level lead to emergent, qualitatively different macroscopic behavior is a fundamental question.

This inquiry benefits greatly from models capable of such emergence. Cellular automata (CA) are prime candidates for this purpose. CA are simple models of computation where cells organized in a grid update their state according to local interactions/rules. They are remarkable for their ability to give rise to complex patterns from these simple rules[4] and some have even been proven to be Universal Turing Machines, that is, machines capable of simulating any other machine.

However, while CA are *capable* of emergent complexity, this is not the norm. Randomly chosen interaction rules often lead to simple or repetitive behaviors. Finding rules that exhibit complexity for longer periods in time and space is surprisingly challenging, and ‘interesting’ rules and initial conditions are highly valued, collected and shared like rare specimens in the community. This raises the question: can we design a CA whose rule self-organizes to these ’interesting’ regimes?

To achieve self-organizing and adaptive capabilities, we introduce heterogeneity into the CA’s interaction rule, a departure from traditional CA principles. This modification allows local neighbourhoods to have different interaction rules based on their history, enabling systems to self-organize from arbitrary initial states to some desired regimes. This publication presents methods for efficiently running these heterogeneous CA and applies them to achieve classical balanced neural networks with excitatory and inhibitory populations as well as self-organized criticality in Ising models.

4.3.2 Results

Efficient computation of heterogeneous CA: To incorporate heterogeneous interaction rules, we generalize convolutions to generic sliding-window operations. By converting our CA grid and its update rule into local patches, we can parameterize the rule locally and define dynamics

[4] The Game of Life (GoL) cellular automata popularized the capability of CA models to exhibit life-like complexity.

on the rule itself with patch-wise operations. This approach significantly enhances the expressivity of CA models with minimal impact on computational costs.

Self-organized criticality: Using both local and global adaptation mechanisms, Ising models initialized arbitrarily were guided to a critical state, as evidenced by measurements of their thermodynamic properties. Local methods achieved the desired states faster and were more robust to a range of initial conditions compared to global methods.

Synaptic plasticity in neural CA: Demonstrating the application of heterogeneity, we simulated large-scale plastic neural CA, efficiently visualizing millions of plastic neurons at high frame rates. Simple plasticity rules enabled these initially random networks to self-organize into an asynchronous irregular state, a common dynamical regime found in spiking networks.

4.3.3 Discussion

This research introduces adaptive CA as a significant advancement, (or cardinality leap) in modeling complex systems. By allowing heterogeneity in the update rules, these models more accurately represent natural systems with feedback mechanisms, where this feedback is key to self-organizing systems into desired dynamical states. The enhanced expressivity of adaptive CA, coupled with their efficiency, allows for real-time modeling of large-scale systems—a vital aspect for prototyping and understanding emergent behavior and self-organization in complex systems. The findings suggest a broader applicability of these models, extending beyond the specific systems studied to any complex system that requires heterogeneity or feedback.

4.3.4 Contributions

- **Sina Khajehabdollahi** was responsible for writing and running the code in this project as well as the writing and editing of the paper.
- **Emmanouil Giannakakis** contributed to defining (and debugging) the plastic neural CA models as well as the writing and editing of the paper.
- **Victor Buendía** contributed to the discussion and formulation of the self-organizing mechanisms for the Ising model.
- **All authors:** contributed to the discussion and review process of the paper.

4.4 Environmental variability and network structure determine the optimal plasticity mechanisms in embodied agents

Giannakakis, E., **Khajehabdollahi, S.**, Levina, A. (2024). Environmental variability and network structure determine the optimal plasticity mechanisms in embodied agents. Artificial Life Conference Proceedings 35. Vol. 2023. No. 1. doi:10.1162/isal_a_00606. arXiv:2303.06734.

4.4.1 Summary

How does neural plasticity emerge to deal with variable environments? How does network structure or task complexity constrain plasticity rules? Living organisms display a remarkable ability to learn and adapt to an uncertain and open-ended world. We hypothesize that such lifelong learning evolves out of the necessity to constantly adapt to variability, when it is impossible to hard-code optimal behaviour. However, it has also been shown that there is strong redundancy in the number of plasticity mechanisms that can achieve identical dynamics in models of brain circuits [134], a result which complements our findings when embedding plastic networks within larger networks. To further understand the relationships between environmental variability, network structure and plasticity rules, we run a variety of evolutionary simulations to discover optimal plasticity rules in different conditions. Two types of agents are defined, a static and an embodied moving agent. In both models, agents are tasked with making either precise predictions (value of the food presented) or binary decisions (eat/don't eat) about a stimulus they receive (a feature vector representing the ingredients of food). In the embodied model, the agents must also navigate a 2D grid where the food is distributed randomly.

We interpret the rules of evolved/optimized agents to compare the different discovered solutions for each condition. The evolved parameters (θ) of each agent represent coefficients of a generalized reward-modulated hebbian plasticity rule[5] [135–137]. This parameter structure allows us to interpret the evolved rules in terms that are linear combinations of the input, output, and reward at each time step:

$$\Delta W_t = \eta_p[R_t \cdot \overbrace{(\theta_1 X_t y_t + \theta_2 y_t + \theta_3 X_t + \theta_4)}^{\text{Reward Modulated}} + \underbrace{(\theta_5 X_t y_t + \theta_6 y_t + \theta_7 X_t + \theta_8)}_{\text{Hebbian}}]. \quad (4.1)$$

4.4.2 Results

Impact of Environmental and Reward Variability on Learning Rate: Simple 'static' agents tasked with identifying the value of presented food, without the need to navigate, demonstrate that the evolved learning rate is directly influenced by environmental variability factors. These factors

[5] Reward-modulated plasticity rules are candidate models in reinforcement learning to adapt neural circuits in behaviourally relevant ways via **local** synaptic interactions as opposed to back-propagation.

include the distance between environments (characterized by the distance between the vector of values for a given ingredient/feature in food), the variability/uncertainty of rewards, and the frequency of environmental change. The evolved learning rate grows with the distance between the environments and decreases with the reward variance. The frequency with which environments change has a non-monotonic relationship with the learning rate showing two different solution strategies: adapting quickly and precisely to sparsely changing environments, and adapting slowly and more broadly to fast changing environments. Similar results were found for the moving agents with small differences.

Evolution of Learning Rules - Decision vs. Prediction: The form of the evolved learning rule are notably different in both static and moving agents based on the task type imposed. In both tasks, the optimal rules exhibit similar behaviours though they converge to different rules. In the prediction task, where the network must output an accurate scalar value representing the ground truth value of the food, the evolved plasticity rule ΔW_t behaves such that it converges to a mean 0 gaussian centered on the ground truth. In the decision task where the ground truth values are obfuscated by the step nonlinearity, a different form of the rule evolves such that ΔW_t diverges in absolute value that are nonetheless reward-correlated.

Both plasticity rules were of a Hebbian form, coordinating between the pre-synaptic activity (the sensory neurons X_t) and the post-synaptic activity (the output of the network y_t), modulated by the reward signal R_t as a threshold. Moving agents also exhibit distinct problem-solving approaches between tasks, with decision tasks showing a stronger evolutionary bias toward a unified rule, contrasting the variety of equally successful yet degenerate rules in prediction tasks. This highlights how the details and feedback from a task critically shape the optimization and evolution of the learning rules that emerge to solve these tasks.

Plasticity Rule Diversity in Embodied Agents: A key finding in the study is the diversity of plasticity functions available to the moving agents compared to static agents. For moving agents, plasticity does not need to precisely decode environmental signals but instead generate interpretable outputs for the motor network which must integrate this signal to navigate the agent and make decisions. Consequently, a broad spectrum of successful learning rules emerge across runs in contrast to the strong convergence to the same rule observed for the static agents that lack a motor network. This trend is observable in both decision and prediction tasks.

4.4.3 Discussion

This research emphasizes the significant impact of environmental and structural variability on the emergence of neuronal plasticity mechanisms in artificial agents. A heterogeneous environment, a reliable sensory system, and balanced rates of environmental change are key for effective synaptic plasticity adaptation. These findings are extended to embodied agents engaged in foraging tasks, demonstrating that environmental dynamics similarly promote plasticity development. The interplay among different network elements, such as motor and sensory networks, can allow for a spectrum of successful learning algorithms. The findings suggest that the function of plasticity rules cannot be studied in isolation but must be considered in conjunction with the connectivity and topology of the networks they operate within as well as the form of the output signal/task. Overall, the research presents synaptic plasticity as a versatile tool adaptable to different environments and tasks, offering insights for both understanding biological learning processes and developing more adaptive and autonomous artificial systems.

4.4.4 Contributions

- **Sina Khajehabdollahi** contributed to the project formulation, particularly of the embodied/moving agents, code base, analysis of results and figures, as well as writing, editing and reviewing the text.
- **Emmanouil Giannakakis** contributed to the project formulation and design, code base for static agents, analysis of results, as well as writing, editing and reviewing the text.
- **Anna Levina** contributed to the project discussions as well as writing, editing and reviewing the text.

4.5 Emergent mechanisms for long timescales depend on training curriculum and affect performance in memory tasks

Khajehabdollahi, S., Zeraati, R., Giannakakis, E., Schäfer, T., Martius, G., Levina, A. (2024). Emergent mechanisms for long timescales depend on training curriculum and affect performance in memory tasks. The Twelfth International Conference on Learning Representations (ICLR 2024). OpenReview:xwKt6bUkXj (accepted). arXiv:2309.12927.

4.5.1 Summary

How do recurrent neural networks (RNNs) solve tasks with long timescales? What mechanisms give rise to long timescales and are there optimal strategies?

Neural networks, biological as well as *in silico*, are adept at exhibiting complex, goal-oriented planning and behaviour with extended temporal dependencies[6]. Tasks with temporal dependencies often demand the network to hold information in memory or for information in distant periods of time to interact. To accomplish this, a network must find ways to maintain long-range correlations to allow the possibility of temporally distant interactions, where the mechanisms by which these long-range correlations emerge are still not fully understood. The timescales of individual neurons are determined by both their biophysiology (for example by its membrane time constant), as well as from network mediated timescales, determined by the activity of neighbouring neurons. Understanding how the interplay between individual neuron properties and network-mediated activity can coordinate macroscopic behaviour with long temporal dependencies is vital for both the design and optimization of RNNs in machine learning and to better understand the possible mechanisms that are at play in biological neural networks. To this end, this study delineates the roles of single-neuron timescales (τ), network-mediated single-neuron timescales (τ_{net}), and collective population timescales (τ_{pop}) in memory-dependent tasks, addressing a gap in the current understanding of these mechanisms.

4.5.2 Results

Necessity of Curriculum: Training RNNs with a curriculum is necessary to solve tasks with long memory requirements when solving the N-Parity task or the N-DMS (delayed match-to-sample) task[7]. RNNs trained without a curriculum increasingly struggle to solve long memory tasks ($N > 10$) as N grows. For larger N, more training time often leads to catastrophic forgetting or requires finely tuned schedulers to anneal the learning rate to ensure progress. We find that introducing either a single-head, or better yet a multi-head curriculum significantly enhances the network's capacity to learn tasks with larger N. The single-head curriculum starts with a single, easy to solve task with short timescales ($N = 2$) and progressively makes the task more difficult ($N \leftarrow N + 1$)

[6] Examples include: sensory perception, motor functionality (walking, dancing, swimming), working memory, language (reading, writing, speech), music, goal-setting, planning.

[7] The value of N determines the number of timesteps the model must keep in memory to solve the N-Parity or N-DMS task.

once it achieves 98%+ accuracy. The multi-head curriculum starts the same way, but whenever a task is solved for a particular N, a *new* readout head is added to the network to solve the task for $N + 1$ while all the previous readout heads must continue solving the task for their respective Ns.

Comparison of Curriculum Performance: Networks trained with a mutli-head curriculum out-perform networks trained with a single-head curriculum, and both are better than no curriculum. The single-head curriculum enables the RNNs to solve tasks up to $N \approx 35$ and $N \approx 90$ for the N-Parity and N-DMS task respectively, whereas the multi-head curriculum can solve for $N \geq 101$ for both tasks while requiring less training data and compute time. The single-head curriculum networks suffer from catastrophic forgetting as N increases, whereas the multi-head networks do not while also exhibiting much more stable training dynamics.

Necessity of Training τ: The time-constant (τ_i) of each neuron i determines how slowly the state of a neuron changes per timestep, where larger τ result in more slowly changing neurons. In single-head curricula, training individual neuron τs is crucial for performance, while multi-head networks perform equally well with fixed $\tau = 1$, suggesting they use other mechanisms beyond single-neuron timescales for memory retention.

Mechanisms Underlying Long Timescales: The study demonstrates that the emergence of longer timescales in RNNs is curriculum-dependent. Single-head networks rely on increasing τ as N increases, whereas multi-head networks leverage recurrent interactions, leading to more effective learning and task-solving capabilities.

4.5.3 Discussion

This study illustrates that training RNNs with curricula that incrementally increase the memory requirements of the task can lead to more effective and robust networks which operate with much longer timescales. Multi-head curricula, which prevent catastrophic forgetting and leverage network interactions rather than individual neuron properties, result in networks that are quicker to train and generalize better to tasks beyond the training set. These findings point towards a more systematic approach to RNN training, favoring the development of longer timescales through network dynamics rather than individual neuron adjustments. Contrary to previous work on the utility of heterogeneity of neuron timescales in network dynamics [138–140], we show that such utility can be contingent on the particular curriculum a network is utilizing during training.

4.5.4 Contributions

- **Sina Khajehabdollahi** contributed to the code base: writing the model, the curriculum learning, a variety of experiments with retraining; writing, editing, and figure creation for the text.
- **Roxana Zeraati** contributed to the project formulation, the code base: timescale estimations and analysis; writing, editing, and figure creation for the text.

- **Emmanouil Giannakakis** contributed to the project formulation, the connectivity analysis, running simulations, writing, editing, and figure creation for the text.
- **Tim Schäfer** contributed to the code base: continuous models, perturbation analysis, a variety of experiments on model variants; writing, editing, and figure creation for the text.
- **All authors:** contributed to the discussion, writing, and review process of the paper.

Conclusions

Conclusions | 5

The projects presented in this dissertation arose from two central questions: **i)** Are there generic or universal properties in collective systems that are privileged for complexity, information processing, adaptivity, and life? **ii)** How can we design and build intuitions of models of self-organization that are more inspired by nature for problem solving?

To answer the first question, our publications in Section 4.1, Section 4.2, and Section 4.5 relate the results of our experiments to criticality through the use of autocorrelation estimations (as is the case for Section 4.1 with image autocorrelations and Section 4.5 with timeseries autocorrelations) or directly by measuring the thermodynamic statistics of spin networks (Section 4.2). These results allow us to make direct statements about the relationship between criticality and performance and optimality within evolutionary, perceptual, and task performance contexts, bridging the gap between theory and application.

The second question is explored in the remaining two publications: in Section 4.3 we demonstrate computational methods to simulate adaptive and heterogeneous CA models at large scales, to accelerate prototyping, intuition building, and interactivity with complex models using a novel implementation that we had yet to see; and in Section 4.4 we demonstrate how degeneracy and optimality in plasticity rules vary as a function of environmental and task parameters, giving novel insights into how plasticity rules are selected for in evolutionary and embodied contexts.

Aesthetics, beauty, and autocorrelations

In the study (Section 4.1 – Assessing Aesthetics of Generated Abstract Images Using Correlation Structure), we observed that some[1] images generated by randomly initialized CPPNs were surprisingly beautiful. This observation led us to ask: **Are there universal statistical properties in images that humans find aesthetically pleasing?** To investigate this, we first parameterized the architecture of the image-generating-model (CPPN), sampling from a broad domain of image statistics as measured by their autocorrelation function. We then generated a corpus of randomly generated images from a variety of architectures and asked 45 subjects to assess their aesthetic appeal. Our findings showed that images deemed aesthetically pleasing by participants had autocorrelation functions that decayed more slowly than the average, although the difference was subtle yet statistically significant. This suggests that even slight variations in the decay rate of autocorrelation functions can influence the perception of beauty, indicating that the perception of aesthetics may be associated with certain statistical properties.

Criticality in evolution

In our second study (Section 4.2 – When to be critical? Performance and evolvability in different regimes of neural Ising agents), we ask

[1] Importantly, only *some* of the generated images were deemed aesthetically pleasing while others were not, which is what originally posed the question, what separates beautiful randomness from the non-beautiful randomness?

the questions: **Do evolutionary systems have attractors near critical regimes? Is this dynamical regime privileged for life, intelligence and complexity?** To answer these questions, we designed experiments with embodied Ising neural networks subject to evolutionary algorithms. We begin by defining our agents as neural networks with Ising interactions to allow measuring thermodynamic properties that display signatures of criticality. We use these measurements to diagnose the dynamical regime of the model throughout its evolution and map its distance to criticality over evolutionary time. The agents must evolve to navigate a 2D environment with simple environmental sensors in order to forage for food, where the most fit agents reproduce in the next generation using an elitism algorithm. We found that agents that were initialized too far from the critical point would struggle to evolve optimal foraging behaviours. On the other hand, agents initialized near the critical point would evolve most rapidly towards their maximum fitness. Notably, we found that given enough time to evolve, agents would not *remain* at a critical point and would instead descend into a sub-critical regime to achieve peak performance. The descent into sub-criticality could be modulated by modifying the task complexity, where more complex tasks remained closer to a critical regime than simpler tasks. We summarize our findings by suggesting that the critical regime provides evolutionary benefits by smoothing out the fitness landscape in parameter space and allowing easier traversal across a broader behavioural space. This benefit becomes less relevant as agents become more fit, after which a sub-critical regime offers more benefits in terms of having more deterministic and structured behaviour once a global minima has already been found. We conclude that initializing a system near its critical points offers benefits by allowing greater evolutionary mobility.

Adaptive & hetergeneous cellular automata

In our third study (Section 4.3 – Locally adaptive cellular automata for goal-oriented self-organization), we aimed to **circumvent a computational bottleneck in modeling complex adaptive systems which are often difficult to scale, slow, non-interactive, or over simplified.** One of the key motivations of this project was to greatly accelerate the speed and interactiveness of these adaptive models in order to build intuitions and heuristics that could ultimately lead to a broader understanding of such local self-organizing models. This thinking is also largely inspired by statements made by Von Neumann in his lectures on theories of self-organizing automata [90] discussing the necessity for heuristics and intuition before laying out mathematical theories. We therefore created a novel implementation of cellular automata that allow for heterogeneous update rules and adaptive dynamics that are both fast and scalable by utilizing convolution-like operations meant for image-processing. Normally, cellular automata are characterized by a rule set that is identical across all cells on its grid which greatly limits the expressiveness of the models that can be written as CA. Here, we introduce a method using patch-wise operations that are similar to convolutions but can be generalized to more complex operations with heterogeneous kernels. We demonstrate the utility of our implementation with two different models: **a)** an adaptive Ising model and **b)** a plastic integrate-and-fire neural spiking model. For the former, it is known that tuning the Ising model to its critical point is non-trivial and requires precise parameter tuning. We then demonstrate that with simple and local adaptivity mechanisms, such a model can self-organize to criticality robustly, even under perturbations or by introducing cells frozen in a particular state. For the latter, it is generally computationally expensive and often impractical to generate large models of spiking networks, especially with plasticity mechanisms, without highly optimized libraries at least in comparison to traditional models of neural networks used in machine learning[2]. We circumvent some of these issues with our implementation by writing our model as a heterogeneous CA (which are by definition local) and demonstrate the ability to simulate millions of plastic neurons with upwards of hundreds of updates per second, allowing for real-time, interactive models to experiment and prototype with much more rapidly than current optimized libraries allow.

Plasticity in evolutionary agents

In our fourth study (Section 4.4 – Environmental variability and network structure determine the optimal plasticity mechanisms in embodied agents), we ask the question: **How do environmental conditions and embodiment affect optimal plasticity rules in neural networks?** This question is motivated by the observation that there is strong degeneracy in the set of plasticity rules that converge on similar statistical outcomes, and so it is not clear how sensitive the landscape of candidate solutions is to actual embodied agents and why certain plasticity rules may be privileged over others in applied settings. To answer these questions, we embed our model of plastic neural networks into agents that must solve a foraging task in order to survive and reproduce in the next generation. We find that the precise details of the optimal plasticity rule are strongly dependent on the details of the task, where the variety of optimal plasticity rules are strongly diminished once specific design choices were made. For example, in the normalization method, environment variability, or whether the agent was embodied and embedded in a 2D space or not. Furthermore, by embedding the plastic network into a larger motor network, we found that we can reintroduce some degeneracy in the optimal plasticity rules as the larger network can co-evolve to interpret the outputs from the plastic sub-network. Overall, we show that optimal plasticity rules are highly sensitive to their precise implementation details and are strongly constrained by these design choices, which is counter to some of the existing work showing degeneracy in plasticity rules.

Emergent timescales in memory tasks

In the fifth and final study (Section 4.5 – Emergent mechanisms for long timescales depend on training curriculum and affect performance in memory tasks), we ask the question: **How do neural networks develop long timescales to solve tasks that require long memory?** This question is motivated by a few observations: **a)** that the brain has a hierarchy of timescales as one traverses into higher-order regions (for example from V1 towards the hippocampus), **b)** that in order to survive, living creatures have to integrate distant memories into actions and plans that can also be extended far forward in time, and **c)** that in the context of the critical

[2] 2: This gap is slowly shrinking [141, 142], however these models are still far from mainstream nor have the same efficiencies or performance.

brain hypothesis, we expect a divergence in autocorrelation timescales[3] in order to readily integrate information across all relevant timescales where the measurements of these timescales act as a proxy to assess the distance to criticality of the trained models.

In order to study how a neural network can maintain such a property, we devised a set of experiments using RNNs that are trained on solving long memory tasks. Training RNNs (or any sequential model) on long memory tasks is a difficult and open problem as current machine learning methods often have issues with exploding or vanishing gradients in time, credit assignment problems, or catastrophic forgetting during training. We found that to ensure stability during training, it was necessary to use curriculum learning methods and that the robustness and performance of our networks were highly sensitive to the details of the curriculum. Furthermore, different mechanisms were found during training depending on the curriculum used, where there was a clear difference in performance depending on which solution was discovered during training. Specifically, when forcing our model to train on multiple tasks concurrently, we found that the mechanism for developing long memories were embedded in the connectivity structure of the network. This was in contrast to models that were trained on one task at a time, whose mechanism for long memories were embedded in the intrinsic timescales of individual neurons. We hypothesize that this multi-task training encourages the discovery of shared and repeated features in similar tasks, giving rise to more universal features that can be reused by different task readout heads. This was an unexpected result as there are numerous previous studies showing how heterogeneous intrinsic timescales in neurons offer strong performance upgrades to networks. While our results do not negate these previous statements, we show that there is a much stronger effect in performance and robustness when slow timescales emerge as a result of network-mediated activity as opposed to intrinsic slowness of neurons.

Overall, we find that depending on how tasks are presented during training and whether or not they are solved myopically or concurrently strongly determines the performance, transfer learning capabilities, and robustness of RNNs training on long memory tasks, where concurrent methods prove vastly superior. Furthermore, task performance was directly related to the ability of the networks to generate increasing timescales which is directly related to the model's distance to criticality [143, 144].

Summary

The projects in this dissertation each explore different elements of complex systems, from their perception by human observers, the possibility of universally optimal dynamical regimes, designing efficient and scalable adaptive models for prototyping and interactivity, the optimality of plasticity rules in embodied neural networks, and the emergence of long timescales in RNNs. Each project seeks to contextualize adaptive systems in more applied and human-interactive contexts in an effort to bridge a gap between theoretical studies of self-organizing models and applied methods for function and performance. In each case, we find that certain universal properties can emerge (*e.g.* power-law acfs, near-critical

[3] For example, in a simple first-order autoregressive model with random noisey input $A(t+1) = \lambda \cdot A(t) + \xi$, the response of the model to activity, (*e.g.* a neural spike) is given by the branching parameter λ. The autocorrelation time of this process is then related by $\lambda^k = e^{-k/\tau}$ where $\tau = -\frac{1}{\log \lambda}$ (where we set the lag $k = 1$) [143, 144]. Thus, as we approach $\lambda = 1$, the point that separates the stable from the unstable regime in the model, the timescale of the autocorrelation diverges. We can use this relation to assess the distance to criticality for data whose underlying dynamical regime is not known. As $\tau \to \infty, \lambda \to 1$.

dynamical regimes) but that significant deviations from these universal properties are strongly characterized by the details of the tasks/contexts we embed these systems in (*e.g.* the degeneracy, or lack-thereof, of plasticity rules, or mechanisms for slow dynamics/long memory). As such, across the projects in this dissertation, we find that universal statistical properties of complex systems can act as initializations for systems, after which fine-tuning is necessary to align these systems to functional or applied settings.

Appendix

A

Publications

A.1 Assessing Aesthetics of Generated Abstract Images Using Correlation Structure

Assessing aesthetics of generated abstract images using correlation structure

Sina Khajehabdollahi
Eberhard Karls University of
Tübingen, Germany
Email: sina.abdollahi@gmail.com

Georg Martius
Max Planck Institute for Intelligent Systems
Tübingen, Germany
Email: georg.martius@tuebingen.mpg.de

Anna Levina
Eberhard Karls University of
Tübingen, Germany
Email: anna.levina@uni-tuebingen.de

***Abstract*—Can we generate abstract aesthetic images without bias from natural or human selected image corpi? Are aesthetic images singled out in their correlation functions? In this paper we give answers to these and more questions. We generate images using compositional pattern-producing networks with random weights and varying architecture. We demonstrate that even with the randomly selected weights the correlation functions remain largely determined by the network architecture. In a controlled experiment, human subjects picked aesthetic images out of a large dataset of all generated images. Statistical analysis reveals that the correlation function is indeed different for aesthetic images.**

***Keywords*—image generation; CPPN; neural network; image statistics; aesthetics; correlations**

I. Introduction

We all marvel at the aesthetic beauty of natural landscapes, the statistics of these images shapes our understanding of beauty. To which extent different statistical measures of the image affect the perception of beauty remains a big question. Photographs of the real world or figurative art mixes the interpretation of the objects and the narrative of the image with one's aesthetic perception. Investigating how generated abstract images are perceived by human observers allows one to uncover the underlying principles.

Natural images follow universal statistical characteristics manifested by specific scaling in the power-spectrum and in the two-point correlations [1]. However, even after two-point correlations are removed, natural images remain recognizable due to higher order relations [2]. Still, two-point correlations affect our perception both in natural and artificial images. Here we aim to understand, on the one side, to which extent such correlations relate to aesthetic attractiveness of images and on the other side, how specific features of artificial neural networks can be used to generate specific statistics of correlations.

The automatic generation of images is typically designed to mimic real scenes, e.g. by attempting to represent the probability density of a large set of images using deep networks and variational inference (Variational Autoencoder [3] or by adversarial training (Generative Adversarial Nets [4]). However, for answering our questions, images generated in this way are not suitable, as they do not allow to disentangle the effects of correlation statistics and concrete object interpretations. Ideally, we need a way to generate abstract images without anchoring them on predefined examples.

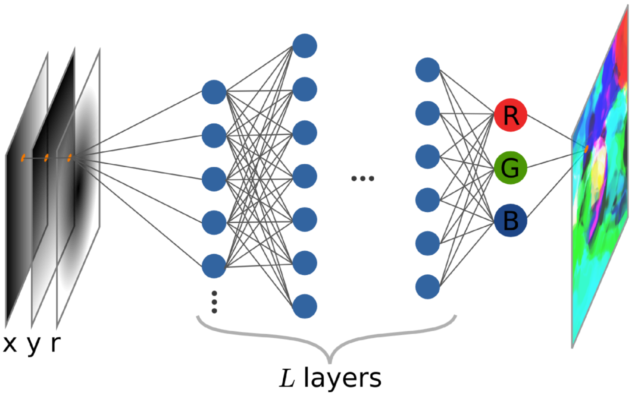


Fig. 1. Illustration of the CPPN architecture. The network processes each pixel independently, here parametrized by the coordinates x, y and radius r. We vary the number of layers, L, and the total number of neurons, N (all blue), and the relative layer sizes.

Compositional Pattern-Producing Networks [5] (CPPNs) are capable of generating a large variety of images without being trained on a particular set of scenes. Images generated by specially trained/evolved CPPNs [6] were shown to be identified as natural objects both by humans as well as deep neural networks [7] albeit being of abstract nature. On the other side, without training or evolutionary selection CPPNs simply transform and decorrelate highly correlated inputs. Interestingly, these neural network with random weights create structured outputs that vary significantly among architectures and hyper-parameters.

We investigate, how constraining the layout of CPPNs in terms of numbers of layers and number of neurons per layer shapes the output correlations. We demonstrate that the ensemble of such networks can generate the breadth of correlation structures similar to the ones observed over different types of natural images. We assess the attractiveness of the generated images to human subjects and evaluate how it is connected with the statistics of two-point correlations.

II. Methods

A. Generating images with CPPNs

We use Compositional Pattern-Producing Networks [5] (CPPNs) with random weights to generate images of size 512 pixels by 512 pixels. The architecture is illustrated in Fig. 1. To explore a variety of image styles, the architectures of the CPPNs are varied. To allow for a controlled generation of architectures we parametrize different aspects with five hyperparameters. $L \in \{3, 5, 10\}$ is the parameter defining the total number of layers in the architecture and $N \in \{100, 250, 500\}$ is the total number of neurons, see also Fig. 1. An initial trial-and-error search suggested that interesting results can be obtained by varying number of neurons in each layer. Thus, the following equation expresses the number of neurons $n(l)$ in layer l as:

$$n(l) = C_N N e^{\mu l/L} \left(\alpha + \sin(-\omega l)\right). \tag{1}$$

$\mu \in \{-1, -0.5, -0.1, 0, 0.1, 0.5, 1\}$ is the decay rate parameter that specifies how the number of neurons decreases (or increases), $\omega \in \{-2, -1, -0.5, 0, 0.5, 1, 2\}$ is the frequency parameter that allows us to introduce bottlenecks to the architecture, $\alpha \in \{2, 5\}$ is a parameter that controls the strength of the oscillation terms, and C_N is a normalizing factor that ensures the total number of neurons in the architecture is as close as possible to N while making sure that no layers have less than 2 neurons (which often results in a simple, solid coloured image).

1) Network details: The layers are fully connected and each neuron also has a bias weight. Activation functions are hyperbolic tangents with cubed inputs ($f(z) = \tanh(z^3)$) in the hidden layers and a sigmoid for the output layer (for red, green, blue colour channels). Once a network architecture is defined, all weights are sampled from a normal distribution with mean 0 and standard deviation 1. An image is generated by feeding in each pixel independently in form of (x, y, r) where $x \in [-1, 1]$, $y \in [-1, 1]$ are the 2D coordinates (with $(0, 0)$ at the center) and $r = \sqrt{x^2 + y^2}$ is the radius.

Due to the multi-layer structure and random weights the network transforms the simple inputs and creates intricate images. To give an intuition, a particular network might give larger weights to the r component initially, giving rise to images whose structures are more circular and symmetric. Further layers act to distort the image. Stronger weights give rise to noisier shapes and colours and smaller weights to simple colours and shapes which tend to be smoother.

2) Generated image database: We generated a large dataset of 35280 images, 40 images per 882 unique architectures. For each architecture configuration, 40 images are generated by changing the random number generator (RNG) seed. These seeds are then re-used for different architecture configurations to allow for easier comparison.

B. Image analysis

In the following we detail the statistical tools and methods used to analyze both generated images as well as natural images. The main measure of choice is the spatial correlation function which we efficiently calculate using a convolution method. In order to answer our research questions, we asked 45 human participants to select aesthetic images as detailed in Sec. II-B2. The correlation functions of these aesthetic images were then compared to both the full dataset and to more restricted architecture subsets. Furthermore, a set of natural images (photographs of nature, cities, animals, landscapes and the like) are compared to the CPPN-generated dataset based in their correlation function. To compute whether a certain set of images has a significantly different correlation function than another set, we are using a Welch's test (similar to a t-test but for samples of varying size and variance).

1) Correlation function: To calculate the correlation function, the Pearson correlation coefficient Eq. 2 of each image is calculated as a function of pixel-wise distance using only the luminosity information (not taking into account colours). For example, for a distance of 1 pixel, all possible pixel pairs that have a distance of 1 would be put into the two vectors X, Y where the Pearson correlation coefficient is calculated as:

$$\rho_{X,Y} = \frac{\mathrm{E}\left[(X - \mu_X)(Y - \mu_Y)\right]}{\sigma_X \sigma_Y} \tag{2}$$

$$= \frac{\mathrm{E}[XY] - \mathrm{E}[X]\mathrm{E}[Y]}{\sqrt{\mathrm{E}\left[X^2\right] - [\mathrm{E}[X]]^2}\sqrt{\mathrm{E}\left[Y^2\right] - [\mathrm{E}[Y]]^2}} \tag{3}$$

where $\mathrm{E}[X]$ denotes the expectation of random variable X, $\mu_X = \mathrm{E}[X]$ and σ_X are the mean and the standard deviation of X. Note that the larger the distance the fewer pairs exist. Conceptually, we will use the correlation coefficients for a given distance, yielding a correlation function: distance vs. correlation coefficient for each image. In the remaining of this section we elaborate on how to make these computations efficient. Understanding this is not essential and might be skipped.

FFT-convolution methods can be employed to quickly calculate the terms in Eq. 3 without the need to extract pairs explicitly. In the simplest case, one can convolve an image with its horizontally and vertically flipped version (using python numpy syntax): `convolve(image, image[::-1, ::-1])`, to obtain a non-normalized version of a correlation coefficient. However, to obtain the Pearson correlation coefficient, normalization with respect to the number of pixels involved at each distance/angle, sample means and sample variances must also be computed as in the terms in Eq. 3. The number of pixels contributing to the correlation coefficient for a distance of x and y pixels horizontally and vertically is then given by $N(x, y) =$ `convolve(I, I)` where `I` is a matrix of ones the same shape as the image. The sample means are defined as $\mu(x, y) =$ `convolve(image, I)/N(x,y)` where we also compute the flipped version and denote it as $\mu(-x, -y)$. If we let $\vec{a} = (x, y)$ then we can write the correlation coefficient as:

$$\rho(\vec{a}) = \frac{\sum_{\vec{p}=(1,1)}^{(N,N)-\vec{a}} i(\vec{p} + \vec{a}) i(\vec{p}) + N(\vec{a})[-\mu(\vec{a})\mu(-\vec{a})]}{N(\vec{a})\sigma(\vec{a})\sigma(-\vec{a})}, \tag{4}$$

where $i(\vec{p})$ is luminescence of the image at position $\vec{p}$. All terms can be calculated using convolution operations. The output of this method gives a matrix that is twice the resolution of the input image, however due to the symmetry in translations of a and $-a$ we can just take the upper diagonal of this matrix, giving us $((1024 \times 1024) - 1024)/2 = 523776$ correlation coefficients per image. The correlation coefficients obtained now have both a distance and angle to which they correspond to. Typically we average out the angle information yielding a single curve: distance vs. correlation coefficient.

2) Aesthetic selection: To identify 'aesthetic' images in the CPPN dataset we designed the following experiment. We consider subsets of 200 randomly sampled images, two per participant, one coloured and one transformed into grayscale. We recruited 45 participants, each shown a unique pair of of set. Each participant had a maximum time of 10 minutes to go through one set of images (20 minutes total for both sets) where they are instructed to tag any image they deem aesthetic/beautiful/attractive (we gave no strict criteria to the participants inviting them to use the most appropriate synonym) using the "XnView" program [8]. No other criteria was prescribed. Participants could scroll through the set and edit the tags as much as they wanted up to the maximal time, or finish rating earlier. 1403 grayscale images were tagged as aesthetic (from the 9000 grayscale images presented), and 1314 colour images were tagged as aesthetic (again from the 9000 colour images presented). A selection of tagged images is presented in Fig. 7.

3) Natural images: To compare to natural images, we have chosen the McGill Calibrated Colour Image Database [9] and used the sets: Animals, Flowers, Foliage, LandWater, ManMade, amd Snow. These images were scaled down so that their smallest dimension was 512 pixels, and center-cropped so that their final resolution was exactly 512 by 512 pixels.

4) Welch's test: To answer whether the correlation functions for two different image sets are significatly different a one-sided Welch's test was used. For presentation purposes we averaged correlation functions from images generated by the same architecture (see Fig. 3). In the case of the aesthetic set we pool together all the 1403 (1314) of grayscale (coloured) images tagged by the participants. For the natural images, each category has anywhere from 150 up to 1112 images. The correlation functions are then binned into 512 distance bins such that the different orientations and discrete nature of the square images are coalesced. We use a Bonferroni correction for multiple comparison: the threshold of significance is divided by a factor proportional to the number of bins within our scope of interest. When comparing the full datasets these are the bins up to a distance of 300 and when comparing the architecture specific datasets these are the bins up to a distance of 100.

III. Results

A. Influence of network architecture on image statistics

Using the architecture parametrization equation (Eq. 1) 882 unique architectures were defined and 40 images were generated per architecture for a total of 35280 images. Some architectures were found to produce images of similar quality and statistics while others were found to be comparatively unique in their outputs, see Fig. 7. Ultimately the parametrization prescribed in this paper allowed the CPPNs to generate a variety of patterns and forms (see Fig. 2 for examples). Images with many layers tend to have fractal-like patterns with sharp edges and boundaries, as if the images from an architecture with less layers have folded in on themselves multiple times. Images with more neurons tend to have smaller scale patterns and clusters, and are generally noisier. However, the distribution of neurons across the layers is found to be of vital importance as well. For example if progressive layers in a network had a growing number of neurons, the resulting images it generates tends to have much smoother and simpler shapes than an architecture with the same number of layers and total neurons but with more neurons in its initial layers and decreasing neurons in progressively deeper layers. The exponential growth/decay and periodic oscillation terms in the architecture parametrization Eq. 1 allow for the creation of a variety of patterns for the same number of layers and neurons.

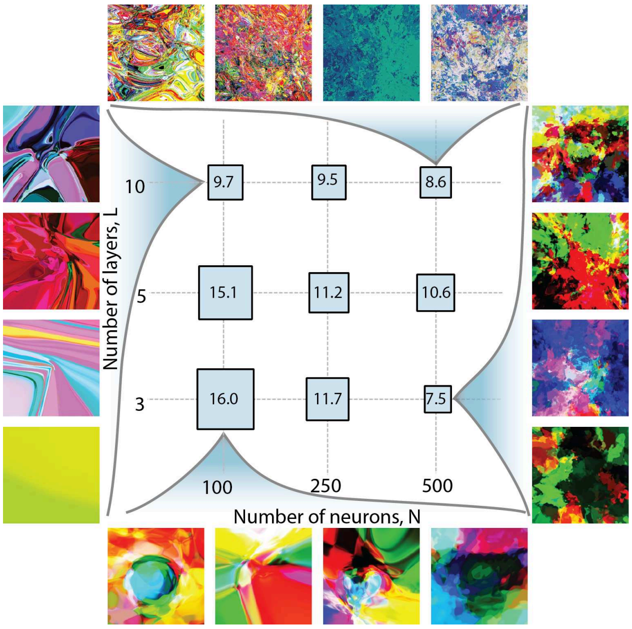


Fig. 2. **Diversity of images generated by CPPNs and distribution of aesthetic images across architectures**. All possible architectures are grouped according to their number of neurons and number of layers (N, L) parameters. Example images (not necessarily tagged as aesthetic) are shown for 4 of these classes. The sides of squares in the central panel are proportional to percentage of the aesthetically tagged images that belong to this class out of all coloured tagged images (the percentage is indicated in the middle of the square). The largest class, $(N, L) = (100, 3)$ contains 210 of 1314 tagged images.

The decay of spatial correlations can characterize the presence of particular scales or scale-freeness [10]. We investigate the spatial correlation of all CPPN images. We average correlations of all 40 images of each network architecture and then plot all of the correlation functions together, see Fig. 3. To understand how particular parameters used for the architecture generation impact the resulting correlations we colour-coded these correlation functions according to the number of layers (L) in their architectures. The clustering of the curves into colour-bands indicates that the number of layers is a dictating parameter. Larger numbers of layers generally results in faster

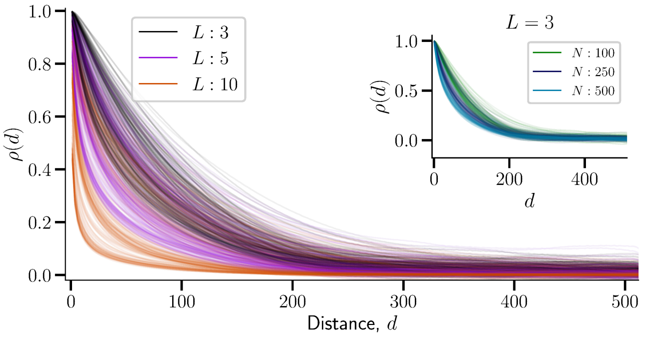


Fig. 3. **Correlation functions of CPPN generated images averaged across architectures.** Each line corresponds to the mean correlation function of 40 images generated for each of the 882 architectures. The lines are then colour-coded according to the number of layers in that architecture. **Inset:** The correlation functions for all architectures with $L = 3$ layers are plotted and colour-coded according to the number of neurons in the network.

correlation decays. However, the curves begin to overlap heavily in region between the fastest and slowest decaying curves. The second most important parameter appeared to be the total number of neurons (N). For any given number of layers L, larger N results in faster decaying correlation functions. For larger L there is larger separation between groups of curves for different N. However, even for the smallest $L = 3$ curves corresponding to the different N are well-clustered as can be seen in the inset of Fig. 3. The other parameters in our parametrization, (α, μ, ω) also played an important role in the quality of the images produced, however, when analyzing the ensemble of image correlations, they were found to not be as important in characterizing their respective ensemble correlation functions.

B. Features of aesthetic images

We investigate the sets of 1403 (1314) grayscale (color) images that were tagged as 'aesthetic' and compared it to the rest of the generated images. Similar to a previous analysis, the spatial correlation functions of the positively tagged images were calculated. We find that images with correlation functions that decay more slowly tended to be picked slightly more often. We investigate this statement statistically using a one-sided Welch's test to compare the means of these sets of images against each other. A significant difference ($p \ll 4 \cdot 10^{-4}$ after Bonferroni correction for multiple comparisons) is found between the set of aesthetic images and the set of all CPPN images for distances up to 97 pixels as seen in the inset of Fig. 5.

In general, such significant difference might be explained if relatively simpler architectures that have a slower than average decay of correlation functions are selected more often. Indeed, we observe in Fig. 4 that the architecture configurations selected most often among both the grayscale images and the colour images tended to trend toward fewer layers and fewer neurons, respectively. These simpler architectures generally generated images with slower correlation decays than the architectures with more layers and neurons.

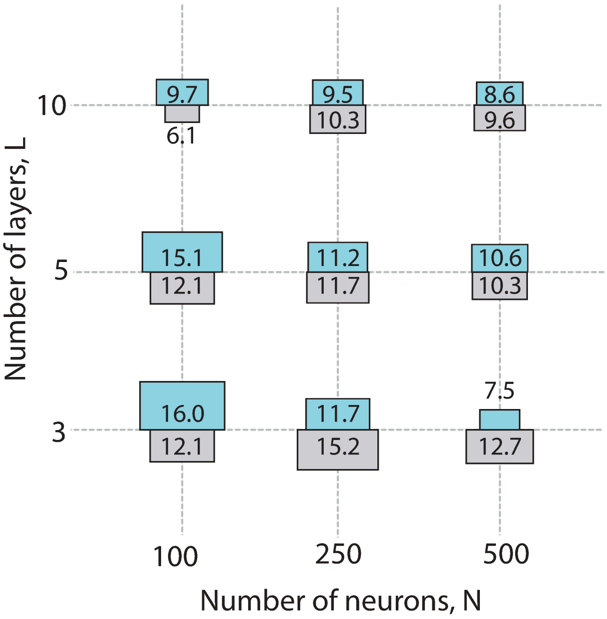


Fig. 4. **Distribution of aesthetic grayscale and colour images across different network architectures**. Grey boxes represent the grayscale images, blue boxes represent the colour images. The sides of the half-squares are proportional to the number of times an image of that class was tagged as aesthetic out of a total of 1403/1314 for grayscale/colour images respectively. The numbers inside the boxes represent the percentage of images tagged within that configuration. All possible architectures are grouped according to their number of neurons and number of layers (N, L) parameters.

Notably, participants were more willing to tag images from more complex architectures with more neurons when judging grayscale images than in colour images. Participants reported that the colour images felt more chaotic than the grayscale and would opt for simpler, smoother structures. Interestingly, while the distribution of aesthetic architecture configurations varied somewhat dramatically between the colour set and the grayscale set, mean correlation functions between these two aesthetic image sets did not show significant differences. This implies that the qualitative differences between these architectures may not show up when comparing their mean correlation functions.

To check if the slower-decaying trends observed globally among the aesthetic images were also observable *within* architectures, we compared the aesthetic images of a particular (N, L) architecture class with the larger population of images within that class. In the same way as before, we apply a Welch's test to compare the correlation functions between the sets and recorded a Bonferroni corrected p-value. By permuting all combinations of N and L, 9 such comparisons can be made for both colour and grayscale image sets. Overall, no obviously significant trend was found between these sub-sets of aesthetic images and their respective architecture classes. This implies that the global behaviour of the entire set of aesthetic images is most strongly affected by the bias towards simpler architectures than any potential exceptionalism of images within their architectures.

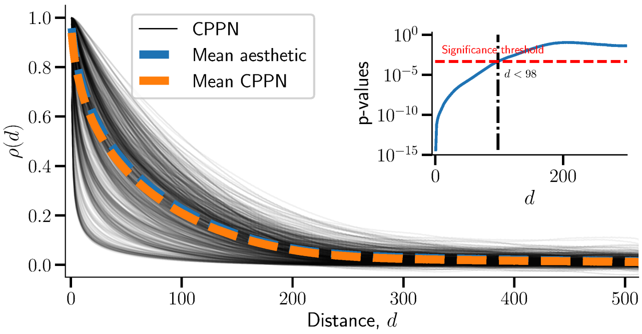


Fig. 5. **Mean correlation functions of the full aesthetic image set and the full CPPN image set.** All CPPN architecture correlation functions (thin dark lines), the means of all CPPN correlation functions (thick dashed orange line), and the means of all the aesthetic images (thick blue dashed line) are plotted in the main figure. **Inset:** The p-values for a one-sided Welch's test (for $d < 300$) between the means of the aesthetic set and the full CPPN set are plotted on a logarithmic scale. The dashed line shows the Bonferonni-corrected threshold of significance at $p \ll 4 \cdot 10^{-4}$ revealing that the sets significantly differ for correlations $d < 98$.

C. Comparison between generated images and natural images

Natural images statistics is influenced by many different parameters, including zoom-levels, scene composition, and so forth [11–13]. Consequently, natural images also posses a broad range of different correlation functions. To test how well the CPPN architectures parametrization could generate a broad space of correlation functions and image structures, we compare its statistics with different sets of natural images, see Fig. 6. Unsurprisingly, different categories of natural images produce a variety of correlation functions. For example the category 'LandWater' have smaller correlations at shorter distances and larger correlations at longer distances and is not particularly well represented by any of the CPPN architectures although images with similar statistics are generated. Meanwhile the categories of 'Foliage', 'Flowers', and 'Animals' have faster decaying correlation functions and were comparable to the slightly noisier CPPN architectures with 5 or 10 layers. In general however, our parametrization seems capable of generating images from a broad space of statistics, at least when compared to these image sets.

A common perception reported by participants was the sense of aesthetics in abstractions arising from the interpretation and identification of familiar objects. Perhaps the aesthetic CPPN images can elicit reactions that resemble those to natural stimuli due to their statistical similarity. To explore in this direction, natural image statistics are also compared to the aesthetic image sets which are categorized according to their (N, L) architecture parameters. These correlation functions are shown on top of the full set of CPPN correlation functions in Fig. 6. It can be seen in this figure that across the bandwidth of correlation functions that are generated by our CPPN architectures, that the characteristic correlation functions of the natural image datasets can be represented by some architectures in our parametrization. The similar correlation functions of the $(N, L) = (250, 10)$ CPPN architecture are compared to the 'ManMade' and 'Snow' natural image set with some example images from each set (the 'Snow' correlation function overlapped almost identically with the 'ManMade' and was omitted for visual clarity). Though the correlation functions of these sets are very similar, it is easy to see that individual images are still very much different. However, some similarities in the structure and distribution of patterns and shapes in the sets can still be observed (see the panels on the side of Fig. 6).

D. Limitations

To extract an elusive 'aesthetic' signal from the noise and variety of human perception and subjectivity is a difficult task. Controlling for human variability and subjectivity demands that we have an understanding of the scope and scale of human variability in this abstract perceptual space, something we do not have. With only 45 subjects, a notion of perceptual clusters began to emerge where some distinct groups of individuals with somewhat similar tastes. However, we still could not look at this sparsely populated space as an ensemble in the same way we could within our neural network architecture space. Gathering data from social media where millions of people around the world voice their opinion is one method that has been employed in the past to alleviate this problem [14]. However, tracking an individual's perceptual space can still be difficult online due to its potentially anonymous nature.

IV. Conclusion and Discussion

In the present work, we explore the capacity of randomly generated CPPNs to generate aesthetic abstract imagery and quantify them using their two-point auto-correlation. We analyse which network architectures with random weights likely generate such statistics. By comparing images that were deemed aesthetic by our volunteer participants we find that in general aesthetic images tend to have correlation functions that decay slightly slower. This result holds true globally but not locally, within individual architecture classes.

It is worth mentioning that any deviations from the mean correlation functions of the full dataset is highly dependent on the parametrization we employ in our architecture sampling equation. If the set of architectures are equally balanced with 'simple' and 'complex' images in perceptual space, then averaging over the correlations of aesthetic images may yield the same global average across all images. In contrast, had we selected more complex architectures which produce noisier images, we might observe that the mean correlations of the aesthetic images would be drastically different from those of the full dataset. When comparing the generated images to natural images we find that our architecture parametrization is capable of representing statistics similar to a variety of natural images. Some participants reported seeing objects, animals, cars or faces in some images. Finding familiarity in such abstractions can itself be a source of aesthetic appeal, at least for a period of time until the novelty wears off [15, 16]. Generally speaking, the battle between order and complexity,

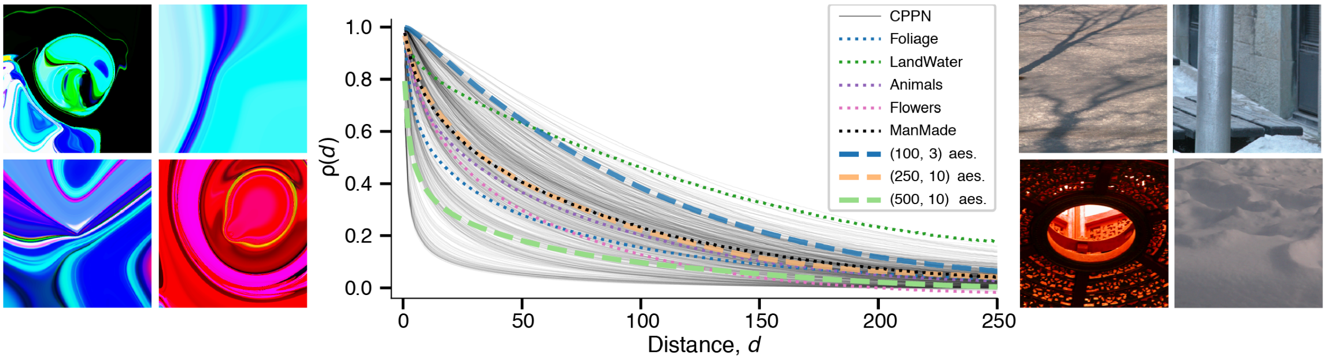


Fig. 6. **Comparison of natural images and CPPN generated images.** Central panel: decay of the two-point correlation as function of distance in different classes of natural images (see legend) and in the groups of aesthetic images corresponding to a particular (N, L) pairs. Left panel: examples of tagged aesthetic images from the $(250, 10)$ group (that corresponds to an orange line in the central panel). Right panel: examples of natural images from 'ManMade' group (corresponds to the black-dotted line in the central panel) and 'Snow' group (omitted for clarity due to overlapping with the 'ManMade' curve).

'unity in variety', is an often discussed topic seemingly at the heart of the emergence of the perception of aesthetics [17].

This study does not explicitly check for power-laws in the Fourier spectrum of the images (which are commonly associated with natural image statistics [1] and are related to the correlation function under some assumptions by the Wiener-Khinchin theorem). We observe, however, in the spatial domain of correlations that aesthetic images tend to have lower decay rates than other CPPN generated images. This long tail of correlations may be rooted in self-similarity which has been observed to be strongly correlated with high aesthetic scores in experiments analyzing the aesthetic scores of artistic imagery [18–21].

Earlier, multiple measures to evaluate aesthetic features of photographs and images were studied in the contexts of automatic image selection [22–24], or evolutionary generation of aesthetic images [25]. Human subjects were shown to be very sensitive to the statistics of local textures [26]. Here we demonstrate that humans prefer images with large correlations at scales exceeding small textured segments.

It can be understood how CPPNs are able to generate such smooth and structured imagery by virtue of the correlation between the inputs that flow independently through the random network. Even though the weights of the network are generated randomly, the inherent correlations in the input can still survive. Smooth/coherent noise has long been used in computer graphics to generate realistic textures and visuals, for example in Perlin noise [27, 28]. Interestingly, CPPNs seem to reverse the order of operations. Instead of starting from a noisy grid and smoothing out the points in-between, CPPNs start with smooth images and progressively add randomness (by random projections and non-linearities) layer by layer. Thus, the underlying transformation is not overly complicated, transforming boring smoothness into structured complexity. By creating similar statistical features as natural images it creates familiar statistics, order, and patterns from pseudo-randomness which humans tend to find delightful [16].

Even though the randomness in the CPPN weights suggests that one has little control over the output of the network, in reality the architecture narrows the type of the picture quite strongly. For example having the final layer with only 1 neuron ensures monochrome images; 3 neurons can be red, green, blue; 4 neurons can be red, green blue, alpha; having a bottleneck layer of only a few neurons in the middle of the architecture can act to simplify the colour palette of the final image; having many layers with a small number of neurons can generate fractal-like images. A variety of interesting architectures capable of generating extremely unique image styles were not included in this project as we opted to study a relatively simple architecture parametrization to keep our variables tractable. Many studies related to weight-agnostic neural networks have found the predisposed biases inherent to particular architectures allowing some of them to perform well without training [29, 30].

We believe CPPNs are an excellent source of unbiased abstract images that may help our understanding on how we define aesthetics and its mathematical boundaries.

Acknowledgment

The authors would like to thank David Ha (@hardmaru) for his insightful tutorial on CPPNs. AL received funding from a Sofja Kovalevskaja Award from the Alexander von Humboldt Foundation, endowed by the Federal Ministry of Education and Research. We acknowledge support from the BMBF through the Tübingen AI Center (FKZ: 01IS18039B).

References

[1] Daniel L Ruderman et al. "Statistics of natural images: Scaling in the woods". In: *Advances in neural information processing systems*. 1994, pp. 551–558.

[2] Bruno A Olshausen et al. "Emergence of simple-cell receptive field properties by learning a sparse code for natural images". In: *Nature* 381.6583 (1996), p. 607.

[3] Diederik P. Kingma et al. "Auto-Encoding Variational Bayes". In: *CoRR* abs/1312.6114 (2013).

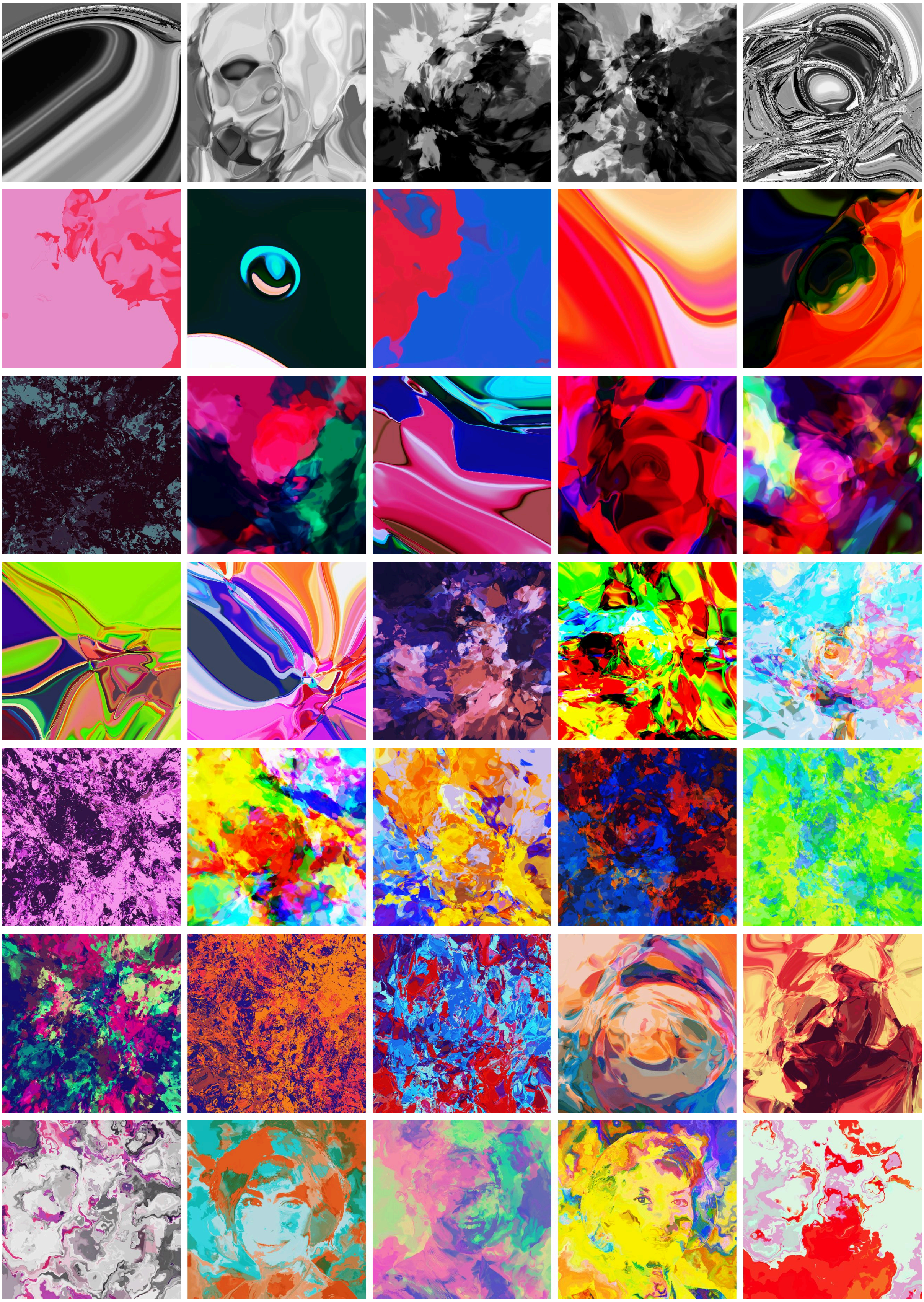


Fig. 7. **Selection of aesthetically tagged CPPN generated images.** Top row shows grayscaled images. The last row is generated adding additional inputs (not used in the experiments). The two outer images are using 2D Perlin noise and the remaining three use pictures of faces as additional input.

[4] Ian Goodfellow et al. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems 27*. Ed. by Z. Ghahramani et al. Curran Associates, Inc., 2014, pp. 2672–2680.

[5] Kenneth O Stanley. "Compositional pattern producing networks: A novel abstraction of development". In: *Genetic programming and evolvable machines* 8.2 (2007), pp. 131–162.

[6] Jimmy Secretan et al. "Picbreeder: evolving pictures collaboratively online". In: *Proceedings of Computer Human Interaction Conference (CHI 2008)*. ACM. 2008, pp. 1759–1768.

[7] Anh Nguyen et al. "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 427–436.

[8] Gougelet Pierre-Emmanuel. *XnView Image viewer*. www.xnview.com. 2019.

[9] Adriana Olmos et al. "A biologically inspired algorithm for the recovery of shading and reflectance images". In: *Perception* 33.12 (2004), pp. 1463–1473.

[10] Andrea Cavagna et al. "Scale-free correlations in starling flocks". In: *Proceedings of the National Academy of Sciences* 107.26 (2010), pp. 11865–11870.

[11] Martin J Wainwright et al. "Scale mixtures of Gaussians and the statistics of natural images". In: *Advances in neural information processing systems*. 2000, pp. 855–861.

[12] Aapo Hyvärinen et al. *Natural image statistics: A probabilistic approach to early computational vision.* Vol. 39. Springer Science & Business Media, 2009.

[13] Holly E Gerhard et al. "Modeling natural image statistics". In: *Biologically Inspired Computer Vision*. Wiley Online Library, 2015, pp. 53–80.

[14] Katharina Schwarz et al. "Will people like your image? learning the aesthetic space". In: *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2018, pp. 2048–2057.

[15] Aaron Hertzmann. "Aesthetics of Neural Network Art". In: *arXiv preprint arXiv:1903.05696* (2019).

[16] Jürgen Schmidhuber. "Formal theory of creativity, fun, and intrinsic motivation (1990–2010)". In: *IEEE Transactions on Autonomous Mental Development* 2.3 (2010), pp. 230–247.

[17] L Neumann et al. "Defining computational aesthetics". In: *Computational aesthetics in graphics, visualization and imaging* (2005), pp. 13–18.

[18] Seyed Ali Amirshahi et al. "PHOG analysis of self-similarity in aesthetic images". In: *Human Vision and Electronic Imaging XVII*. Vol. 8291. International Society for Optics and Photonics. 2012, 82911J.

[19] Seyed Ali Amirshahi et al. "How self-similar are artworks at different levels of spatial resolution?" In: *Proceedings of the Symposium on Computational Aesthetics*. ACM. 2013, pp. 93–100.

[20] Tamara Melmer et al. "From regular text to artistic writing and artworks: Fourier statistics of images with low and high aesthetic appeal". In: *Frontiers in human neuroscience* 7 (2013), p. 106.

[21] Joachim Denzler et al. "Convolutional neural networks as a computational model for the underlying processes of aesthetics perception". In: *European Conference on Computer Vision*. Springer. 2016, pp. 871–887.

[22] Yubin Deng et al. "Image aesthetic assessment: An experimental survey". In: *IEEE Signal Processing Magazine* 34.4 (2017), pp. 80–106.

[23] Ritendra Datta et al. "Studying aesthetics in photographic images using a computational approach". In: *European conference on computer vision*. Springer. 2006, pp. 288–301.

[24] Wenshan Wang et al. "Neural Aesthetic Image Reviewer". In: *arXiv preprint arXiv:1802.10240* (2018).

[25] Eelco den Heijer et al. "Using aesthetic measures to evolve art". In: *IEEE Congress on Evolutionary Computation*. IEEE. 2010, pp. 1–8.

[26] Holly E Gerhard et al. "Towards rigorous study of artistic style: a new psychophysical paradigm". In: *Art & Perception* 2.1-2 (2014), pp. 23–44.

[27] Ken Perlin. "An image synthesizer". In: *ACM Siggraph Computer Graphics* 19.3 (1985), pp. 287–296.

[28] Ares Lagae et al. "A survey of procedural noise functions". In: *Computer Graphics Forum*. Vol. 29. 8. Wiley Online Library. 2010, pp. 2579–2600.

[29] Adam Gaier et al. "Weight Agnostic Neural Networks". In: *arXiv preprint arXiv:1906.04358* (2019).

[30] Dmitry Ulyanov et al. "Deep image prior". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 9446–9454.

A.2 When to be critical? Performance and evolvability in different regimes of neural Ising agents

When to Be Critical? Performance and Evolvability in Different Regimes of Neural Ising Agents

Sina Khajehabdollahi*
University of Tübingen
Department of Computer Science
Max Planck Institute for
Biological Cybernetics
sina.abdollahi@gmail.com

Jan Prosi
University of Tübingen
Department of Computer Science
Max Planck Institute for
Biological Cybernetics

Emmanouil Giannakakis
University of Tübingen
Department of Computer Science
Max Planck Institute for
Biological Cybernetics

Georg Martius
Max Planck Institute for
Intelligent Systems

Anna Levina
University of Tübingen
Department of Computer Science
Max Planck Institute for
Biological Cybernetics
Bernstein Center for Computational
Neuroscience Tübingen

Abstract It has long been hypothesized that operating close to the critical state is beneficial for natural and artificial evolutionary systems. We put this hypothesis to test in a system of evolving foraging agents controlled by neural networks that can adapt the agents' dynamical regime throughout evolution. Surprisingly, we find that all populations that discover solutions evolve to be subcritical. By a resilience analysis, we find that there are still benefits of starting the evolution in the critical regime. Namely, initially critical agents maintain their fitness level under environmental changes (for example, in the lifespan) and degrade gracefully when their genome is perturbed. At the same time, initially subcritical agents, even when evolved to the same fitness, are often inadequate to withstand the changes in the lifespan and degrade catastrophically with genetic perturbations. Furthermore, we find the optimal distance to criticality depends on the task complexity. To test it we introduce a hard task and a simple task: For the hard task, agents evolve closer to criticality, whereas more subcritical solutions are found for the simple task. We verify that our results are independent of the selected evolutionary mechanisms by testing them on two principally different approaches: a genetic algorithm and an evolutionary strategy. In summary, our study suggests that although optimal behaviour in the simple task is obtained in a subcritical regime, initializing near criticality is important to be efficient at finding optimal solutions for new tasks of unknown complexity.

Keywords
Evolutionary optimization, criticality, Ising model, neural networks, dynamical systems

* Corresponding author.

Artificial Life 28: 458–478 (2022) https://doi.org/10.1162/artl_a_00383

1 Introduction

Operating close to the critical point at a second-order phase transition has long been associated with the optimal performance of complex systems. Several biological systems, such as gene regulatory networks (Balleza et al., 2008; Rämö et al., 2006), neural networks (Beggs & Plenz, 2004; Schneidman et al., 2006; Tkačik et al., 2015), collectively behaving cells (De Palo et al., 2017; Halley et al., 2009), swarms (Cavagna et al., 2010; Chaté and Muñoz, 2014), or populations of co-evolving, communicating agents (Hidalgo et al., 2014) have been shown to operate close to a critical point. Criticality has been associated with an ability to solve complex tasks (Villegas et al., 2016), optimal information transmission and sensitivity (Beggs, 2007; Bertschinger & Natschläger, 2004; Boedecker et al., 2012; Kinouchi & Copelli., 2006), flexibility towards changes in the environment, and good evolvability (Aldana et al., 2007) in complex living systems (Kauffman, 1993). In these models, different variations of the transitions and scaling exponents are considered: In the branching model related to neuroscience, it is a transition between absorbing and active state; in the Ising-like models, a transition between ordered and disordered state. However, as long as the model presents a second-order phase transition, most results remain qualitatively unchanged. All these optimized properties provide an adaptive advantage in natural environments, leading to the assumption that evolutionary dynamics push biological systems close to the critical regime.

On the other hand, it has been suggested that the ubiquitous presence of noise in nature pushes living systems into a more robust subcritical regime. For example, in an evolutionary model of random Boolean networks (RBNs), decreasing the system size, making the task less complex, or introducing noise to the system pushes the optimal regime further into the subcritical range (Villegas et al., 2016). Similarly Rämö et al. (2007) observed that whereas information propagation is maximized in critical RBNs, the optimal regime shifts slightly into the subcritical regime under the presence of noise. In recordings from the nervous systems of different animals, slightly subcritical behaviour was observed (Priesemann et al., 2014; Wilting & Priesemann, 2019). A related phenomenon has been observed on neuromorphic chips, which optimally perform in simple tasks when in the subcritical regime, whereas harder tasks require progressively more critical dynamics (Cramer et al., 2020). The escape dynamics of schooling fish remain in subcritical state even under pharmacological manipulation that increases alertness, though more alert fish self-organize closer to critical state (Poel et al., 2022). Finally, for some applications the combination of systems operating at different distances from criticality has been shown to lead to optimal results (Zierenberg et al., 2020). The disordered supercritical state has been universally observed to perform poorly (Kauffman, 1993; Villegas et al., 2016). There are different critical transitions in the examples mentioned above, thus different definitions of what it means to be subcritical. For example, for the order/disorder transition, subcritical refers to the ordered state. At the same time, for the branching network transition to an active state, subcritical means that perturbations are rapidly dying out. Interestingly, the optimal behaviour was found in the subcritical regime for both these transition types.

The benefits of criticality for the evolvability of living systems have been associated with the genotype–phenotype coupling. Specifically, it has been shown (De Jong, 2006) that a tight genotype–phenotype coupling leads to optimal evolvability. Due to this coupling, the dynamical regime has an impact on the properties of the fitness landscape. In an RBN model, the super- and subcritical regimes were shown to disturb the genotype–phenotype coupling (Kauffman, 1993) and lead to either very rugged or overly flat fitness landscapes. A rugged fitness landscape means that the evolutionary dynamics are just a random search and thus inefficient in high dimensions (Kauffman & Levin, 1987). On the other hand, a very flat landscape dampens the optimization process. Both phenomena result in a complexity catastrophe, where an increase of system size leads to a failure to discover satisfying solutions with evolutionary search. Critical RBNs result in intermediately rugged fitness landscapes that allow for efficient hill climbing search and are less prone to the complexity catastrophe.

Our previous study (Prosi et al., 2021), examined how the dynamical regime of populations of evolving organisms influences their ability to solve a task. Our investigation used a simple foraging game of scalable difficulty, where organisms can gain energy by eating food particles and consume energy when moving. We optimized the Ising networks controlling the organisms using a simple genetic algorithm that allowed us to analyze the changes in the dynamical regimes during evolution. In addition, we proposed a potential answer to the question of which dynamical regime demonstrates the best performance and stability with respect to changes in the environment. Still, this study left unclear the extent to which the results are determined by the choice of evolutionary strategy or if they represent a general trend regardless of which evolutionary algorithm is used for optimization. Here, we conduct a more detailed analysis of the dynamics underlying our network model and extend the previous findings by comparing the behaviour of two distinct evolutionary strategies. Overall we confirm that our results are not dependent on the exact algorithm used to train the model.

2 Methods

We investigate a 2D environment where organisms controlled by individual neural networks forage for food. Each organism gains energy by eating food particles and consumes energy by moving. The organisms eat the food particles by running over them and share their environment with other organisms in the same generation. This multi-agent environment is chosen to allow for the environment to complexify as agents evolve and become more adept at their task, thereby changing the distribution of input signals an individual experiences in its lifetime. Furthermore, this type of multi-agent environment forces the agents into a strategic competition with themselves, which again encourages a richer environment. Motivated by the results in Hidalgo et al. (2014) where environment complexity and the optimal dynamical state were positively correlated, we introduce a mechanism to complexify our environment. We can increase the difficulty of the task by requiring the organism's velocity to be below a certain threshold when running over food in order to be able to consume it. The fitness of an organism is determined by its average energy throughout its lifespan. We use two distinct evolutionary algorithms to optimize the network controlling the organisms and compare their performance.

2.1 Organisms

The organisms in our model are controlled by an Ising neural network (INN) that has been previously used in Aguilera and Bedia (2017) as well as Khajehabdollahi and Witkowski (2020). The Ising network consists of N neurons that can be in one of two states $s_i \in \{-1, 1\}, i = 1, \ldots, N$. All neurons are split into three classes: sensory neurons that only receive input from the sensors, motor neurons that control the agent, and hidden units used for computations. Their connectivity is described by the adjacency matrix $A \in \{0, 1\}^{N \times N}$ and the weight matrix $J \in [-2, 2]^{N \times N}$, as shown in Figure 1(b). No connections between sensor and motor neurons are allowed by the adjacency matrix at any time. Following the Ising model, each network activation pattern (vector of states of all neurons) has an associated energy:

$$e(s_1, \ldots, s_N) = -\sum_{i,j} A_{ij} J_{ij} s_i s_j. \tag{1}$$

The network stochastically minimizes the energy by following Glauber dynamics: At each network iteration, all non-sensor neurons are updated in a random order and the state of neuron i changes from s_i to $-s_i$ with probability:

$$p_i = \frac{1}{1 + e^{\beta \cdot \Delta e_i}}, \tag{2}$$
$$\Delta e_i = e(s_1, \ldots, s_i, \ldots, s_N) - e(s_1, \ldots, -s_i, \ldots, s_N),$$

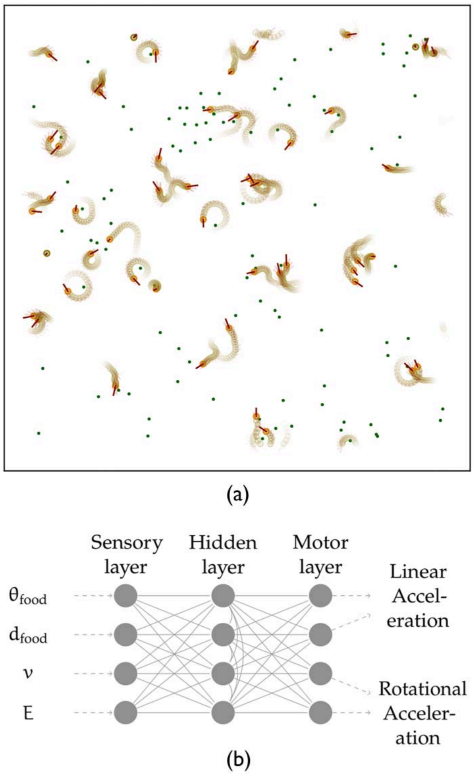


Figure 1. Snapshot of population dynamics and schematic representation of the control network. (a) Environment with 50 organisms (red circles with trails) foraging for food (green dots). (b) A network with 12 neurons: four sensory, four hidden, and four motor neurons. All allowed edges are displayed. The exact topology and edge weights are subject to change by the evolutionary algorithm.

where β is the inverse temperature of the network ($\beta = 1/(T \cdot k_B)$; k_B is the Boltzmann constant, which we set to one and omit for simplicity; and Δe_i is the change in the energy of the network that is caused by the spin-flip of the i_{th} neuron (changing its state s_i to $-s_i$). The energy change Δe_i is determined by the connectivity matrix J and the states of neighboring neurons. A decrease in energy (negative energy change) leads to a greater likelihood of a flip. The parameter $\beta > 0$ controls the likelihood of energetically unfavourable flips. A large $\beta \gg 1$ leads to deterministic network behaviour dominated by the connectivity, whereas a smaller β leads to more random behaviour. For each time step in the simulation, the motor neurons of an agent are read out and apply an action, and the sensory neurons are then updated. The model must then thermalize according to the new values of the sensor neurons by updating its state via Equation 2. In principle, the number of iterations required to converge to equilibrium is a function of the connectivity matrix, the temperature of the model, and the distribution of sensor values. It is known that near the critical point it takes more time to reach an equilibrium state. However, for practical reasons, we fix the number of iterations to 10 thermalization steps. An analysis on the sensitivity of the model to the thermalization time is provided in Appendix 1. In principle this hyper-parameter acts as the amount of time an agent has to "think" about its new sensory inputs, and may be biologically constrained.

An organism has four input neurons that receive information about the angle θ_{food} and distance d_{food} from the closest food particle as well as its own velocity v and energy E (this energy is distinct from the Ising energy e). Moreover, each organism has four output neurons that control linear and rotational acceleration (two neurons each) and N_h hidden neurons (Figure 1(b)). For each time step in the environment we assign a normalized real value to the sensor neurons according to the environmental input. The hidden and motor sensors can only obtain binary states (−1, 1). We

Algorithm 1. Evolution of agents.

```
for generation = 1to Generations do
    Foraging Game in 2D environment
    for $t = 1$to organism's lifespan do
        update sensor neurons(organism)
        for iter = 1to network iterations do                ▷ Glauber update of INN
            for non-sensor neuronin INN do
                potential spin-flip(neuron)
            end for
        end for
        read motor neurons(organism)
        move in 2D environment(organism)
    end for
    Evolve(population of organisms)                         ▷ Evolves $J$, $A$, and $\beta$
        using Genetic Algorithm or Evolution Strategy
    Reset 2D environment
end for
```

equilibrate the hidden and motor neurons for 10 iterations using a Metropolis algorithm (Metropolis et al., 1953) that implements Equation 2 and subsequently reads the states for the motor neurons (see Algorithm 1.). The agent accelerates in case both neurons of a motor unit are in agreement and have positive states, decelerates in case both are in agreement and have negative states, and does nothing if the neurons are in disagreement and have opposite states.

For most simulations we use a hidden layer with 4 neurons, ($N_h = 4$). Additionally, in order to study whether our methods perform well with larger networks, we also simulate a network with $N_h = 20$. At the beginning of each simulation, an organism is provided with an amount of initial energy $E_{\text{init}} = 2$. Movement reduces energy and consuming food particles increases it. We consider two versions of this environment: In the *simple task* organisms consume food when passing over it. In the *hard task* organisms have to slow down and almost stop to be able to consume food. Unless stated differently, a simulation lasts for a *lifespan* of $t = 2{,}000$ time steps after which the evolutionary algorithm is applied, and the task is simple. 50 INN-controlled organisms are placed in a 2D environment with periodic boundaries and ever-respawning food particles, conserved to a value of 100 (Figure 1(a)).

2.2 Evolutionary Algorithms

2.2.1 Genetic Algorithm

The genetic algorithm (GA) applied to the INNs consists of a combination of elitism, mutation, and mating. At the end of the simulation described above, the fitness of each organism is defined as their mean energy throughout their lifespan. Subsequently, the 20 fittest organisms continue unchanged to the next generation; 15 more are added by duplicating the top 10 organisms with a 10% chance of mutations. The remaining 15 are then populated by mating between the current population. The next generation therefore consists of 20 copied organisms, 15 possibly mutated, and 15 generated by mating. The mutation operation adds or deletes edges in A (connections not present in Figure 1(b) cannot be added), re-samples a random edge weight in J from a uniform distribution $\mathcal{U}(-2, 2)$, and perturbs the inverse temperature with multiplicative Gaussian noise $\beta' = \beta \cdot \Delta\beta$, for $\Delta\beta \sim \mathcal{N}(1,\ 0.02)$. Finally, the mating operation randomly chooses two parents from the pool of the 35 individuals that either survived or were mutated duplicates, and takes a weighted average of their connectivities J and inverse temperatures β to produce an offspring. In most of our simulations, the GA iterates for 4,000 generations.

2.2.2 Evolution Strategy

We verify that our results are not contingent on the specific behaviour of the GA described above, by employing an evolution strategy (ES) from the family of natural evolution strategies (Wierstra et al., 2008, 2014) with some modifications. In contrast to the GA these methods parametrize the population by a distribution over the genome and adapt its parameters. The algorithm uses a multi-variate Gaussian distribution $\mathcal{N}(\mathbf{J}, \sigma\mathbb{I})$ with mean $\mathbf{J}$ and fixed variance σ (where $\mathbb{I}$ is the identity matrix). Using a fixed variance simplifies the algorithm and was reported to work well for neural network training (Salimans et al., 2017). The update of the mean $\mathbf{J}$ follows a gradient ascent on the fitness, estimated based on the fitness of n sampled individuals. However, to make the gradient invariant to monotonous fitness transformations, a rank-based fitness is used, as in Wierstra et al. (2014). We start from the implementation by Najarro and Risi (2020) but add elitism in two ways to the algorithm. We compute the gradient with respect to the best individual and keep a small fraction of elite individuals for the next generation. The update of the mean $\mathbf{J}$ is given by:

$$\mathbf{J}_{t+1} = \mathbf{J}_t^\star + \frac{\alpha}{n\sigma}\sum_{i=1}^{n} F\left(\mathbf{J}_t + \sigma\epsilon_i\right)\cdot\epsilon_i', \tag{3}$$

where α is the learning rate, σ is the standard deviation of our Gaussian search distribution, n is the number of individuals generated (in our case equal to the population of the environment), $F(\cdot)$ the ranked fitness (our foraging task), $\epsilon_i = \mathcal{N}(0, 1)$ is a Gaussian random vector, $J^\star$ is the parameter vector of the best individual, and ϵ_i' is the random vector ϵ_i relative to the best, i.e., $\epsilon_i' = J_i + \sigma\epsilon_i - J_i^\star$. The values of F are computed by first ranking all fitness values and then normalizing those ranks by subtracting their mean and dividing by their standard deviation.

We also update the inverse temperature (β) of the model in the same way; however, we let this parameter evolve slower by setting its σ to $\sigma_\beta = 0.1 \cdot \sigma$ in order to keep the ES algorithm comparable to the GA. Further motivation for this choice is that β is a global parameter which should change slowly relative to the connectivity parameters. Furthermore, we do not include any decay rates in the learning rate and standard deviation in order to avoid conflating the dynamics of a decaying learning rule with any convergences that might occur due to selection pressures.

Since each individual per generation is actually competing for the same resources as other individuals (as opposed to running in an independent, parallel simulation), we also employ the use of elitism per generation to ensure that (6/50) previous well-performing individuals are part of the next generation (the ϵs are computed accordingly). Furthermore, to allow for a sparse change in parameters during search we set 50% of entries in ϵ to zero. This parameter was introduced as a variable to control the genetic diversity between generations, again motivated by the idea that individuals are actually competing against one another and not running independently, and therefore should have to compete against similarly evolved individuals.

As is commonly known, the ES is sensitive to the hyper-parameter σ. For example, in the works of Sehnke et al. (2010) considerable work is done to ensure that the σs of different parameters are adaptive, where ensuring that the initial σ_0 is large enough to find a solution. When chosen appropriately we found that our version of the ES optimizes the fitness of populations to values comparable to the GA.

2.3 Defining the Dynamical Regime of an Organism

We use the approximation of the heat capacity from statistical physics to derive a measure of an organism's dynamical regime (sub-, super-, critical). Throughout the article, we define the state of the organisms relative to the order/disorder transition. In our finite system, we estimate the putative divergence point by changing the inverse temperature β multiplying it with a scaling constant c_β. This change of temperature influences how likely the state of the neurons will flip (Equation 2), and thus change the equilibrium distribution of energies e (Equation 1). We search for a c_β that maximizes value of the heat capacity $C_H(c_\beta)$, defined as

$$C_H(c_\beta) = \frac{1}{T^2}\mathrm{Var}(e) = c_\beta^2\,\beta^2\mathrm{Var}(e). \tag{4}$$

We define the $c_\beta^{\mathrm{crit}} = \underset{c_\beta}{\operatorname{argmax}}\, C_H(c_\beta)$. An analogous procedure was used in Tkačik et al. (2015). We then define the distance of the network from the critical point by the logarithm of the scaling factor required to bring the network to criticality, $\delta = \log(c_\beta^{\mathrm{crit}})$. In our case, due to the asymmetric connectivity matrix and non-equilibrium nature of the system, the procedure should be seen as an approximation of the actual heat capacity, resulting in a proxy for critical point. More details can be found in the Results section 3.1. For unevolved organisms whose connectivity matrices are initialized from the uniform distribution $\mathcal{U}(-1, 1)$ (first generation, Figure 2), the relationship between β_{init} and δ can be approximated by

$$\delta \approx \log\frac{1}{\beta_{\mathrm{init}}} = -\log\beta_{\mathrm{init}},\ \beta_{\mathrm{init}} \in [0.1, 10]\,. \tag{5}$$

On a technical note, models in subcritical states can take increasingly large amounts of time to escape from a local optimum into a global optima, which can result in numerical divergences in our estimates of the specific heat $C_H(c_\beta)$. To avoid this issue we employ an annealing method when calculating the specific heat at a given temperature, by first starting at a much higher temperature and then gradually lowering it down to the target temperature. During this process the sensor neurons are kept fixed according to values that the agents had observed and which are saved during training. This method has the benefit of being scalable especially for larger networks, which may have very frustrated connections that take exponentially longer to equilibriate.

During evolution, the distance from the critical point can (and will) change from its initialized state, and we must calculate its specific heat $C(\beta)$ as a function of the temperature scaling parameter c_β in order to find its maximum and obtain an estimated distance to criticality.

3 Results

We perform extensive numerical experiments to investigate the properties of the dynamics of evolving Ising network agents and present the relationships between dynamical states, criticality, and evolutionary fitness. However, before we present results on these dynamical states, we validate that our

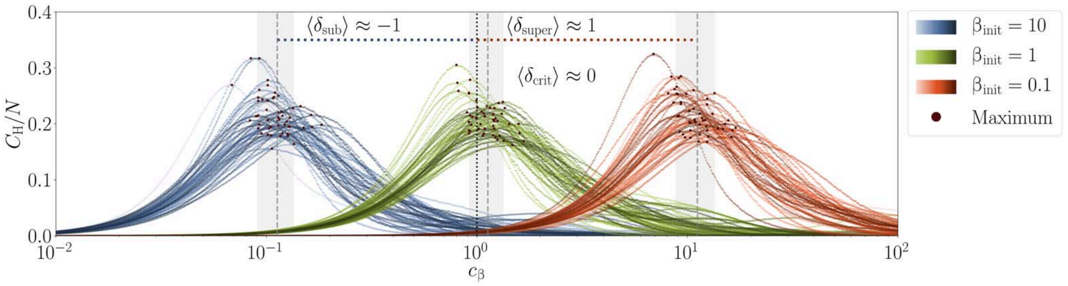


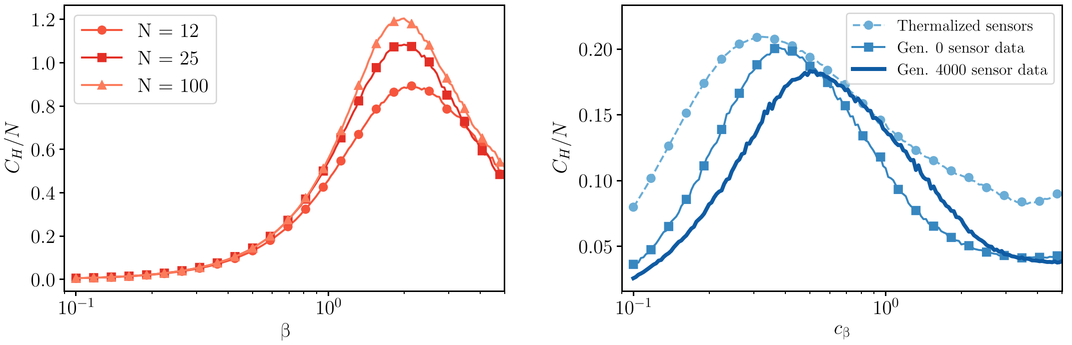
Figure 2. The dynamical regime of a network can be calculated by finding a scaling factor c_β of the inverse temperature that maximizes the heat capacity. Heat capacity (Equation 4) of the Ising networks (Figure 1) for 50 initially subcritical ($\beta_{\mathrm{init}} = 10$, blue), critical ($\beta_{\mathrm{init}} = 1$, green), and supercritical ($\beta_{\mathrm{init}} = 0.1$, red) organisms as a function of c_β. For each organism it reaches the maximum (marked by a dot) at individual values $c_\beta = c_\beta^{\mathrm{crit}}$. Dynamical regime $\delta = \log(c_\beta^{\mathrm{crit}}) \approx -\log(\beta_{\mathrm{init}})$. The displayed populations are unevolved and the resulting dynamical regimes closely correspond to their respective β_{init}.

measures of criticality behave as expected by considering a generalized Ising model that is conceptually in between the classical Ising model and our Ising network agents.

3.1 Criticality in the Generalized Ising Model

While the classical 2D Ising model and its critical point and universality class are well studied, extrapolating these results to different models and particularly to non-equilibrium systems is generally not possible and has to be checked individually for every variation of the model. Particularly, in this article, the controller of the agents is a neural network that represents a generalized Ising model with all-to-all connectivity, as opposed to a regular lattice, and with both positive and negative real-valued weights. Furthermore, the controller neural network receives sensory inputs that perturb the model away from equilibrium, while being at equilibrium is the central requirement for the derivation of the critical points in the 2D Ising model. Finally, controller networks evolve very specific connectivities via the selection pressures of the world/task they are embedded in, and this precludes a scaling analysis on the specific evolved networks to measure if their thermodynamic properties exhibit scale-free, and therefore critical, behaviours. As a compromise, we instead do a scaling analysis on random networks with an architecture similar to our controller networks. We generate ensembles of random networks of sizes $N = 12, 25, 100$, each having one third of its neurons designated as sensor neurons and another third as motor neurons, prohibiting connections between motor and sensor groups. We normalize the weights by the Frobenius norm of the connectivity matrix. We then calculate the heat capacities of these models, where the sensor neurons are given values from the uniform distribution on $[-1, 1]$, shown in Figure 3(a). It can be seen that the specific heat of these models tends to peak around $\beta \approx 1.5$, showing that our normalization captures the changes of the peak location with the system size. The maximum value of the specific heat is growing with system size, implying the existence of a critical point.

The next complication in our methodology is the fact that the sensor neurons perturb the system away from equilibrium at each time step, and therefore make our analysis of criticality more difficult. To better understand the implications of this feature in our model, we can compare how the specific heat of a network changes depending on the statistics of the sensory input and the possibility to thermalize its sensor neurons. We consider the most fit agent from 54 independent simulations of 4,000 generations evolved with the GA. We calculate their specific heats for various inverse temperatures three different ways and average across the agents. In the first method, we thermalize the sensor neurons and treat them identically to the rest of the neurons in the model. This results in



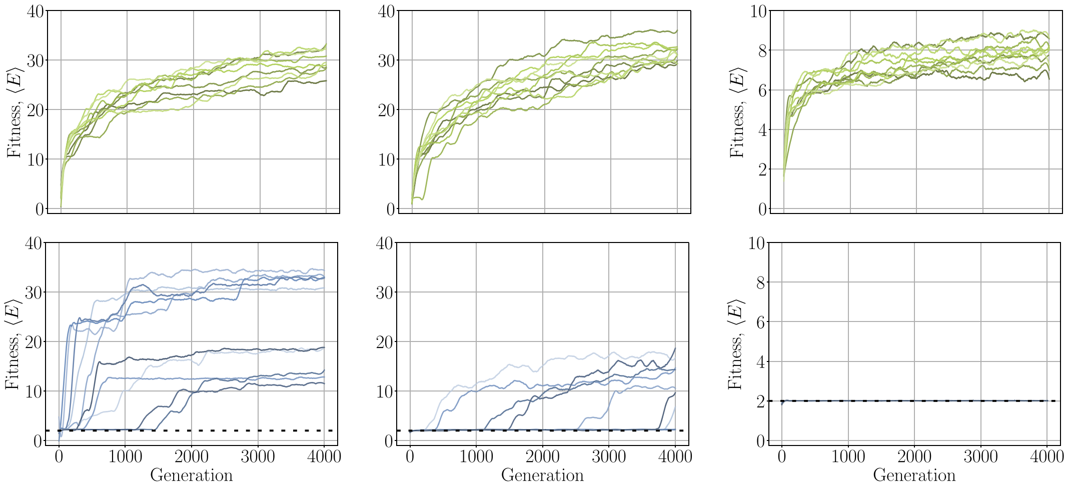
(a) (b)

Figure 3. Numerical indication of the presence of the critical point in our non-equilibrium generalized Ising model. (a) A scaling analysis is done using ensembles of random networks with a similar architecture to the evolving agents. The peaks of the specific heat grow with the size of the system, and its location converges towards a $\beta \approx 1.5$, a property commonly expressed in critical systems. (b) 54 independent, fit agents, evolved for 4,000 generations. The specific heat dependence on the inverse pseudo-temperature with thermalized sensor neurons (equilibrium model, light color), with the sensor data clipped and drawn from the distribution gathered at the 0th generation (darker shade), and with sensor data from the 4,000th generation (darkest line). The peak of the specific heat shifts slightly depending on the method.

the equilibrium model that is different from the Ising model only in the features of its connectivity matrix; see Figure 3(b) lightest curve. We repeat the same calculations with clipped sensor neurons drawn from the distribution of sensor data gathered in its final generation (4,000), which we save throughout the lifetime of the agents. This is the *effective* specific heat of the embodied model as it interacts with its environment, and it is how we defined the state of the agents in the rest of the article; see Figure 3(b) darkest curve. Finally, we calculate the specific heat using the sensor data from the 0th generation, where due to the unevolved state of the agents, the sensor data are less diverse. The difference between the thermalized specific heat and the *effective* specific heat is a slight shift in the location of the peaks. Furthermore, it can be seen that the evolved agents are closer to their maximal susceptibility when their specific heat is calculated from the environment they are actually embedded in. In other words, if we were to calculate the specific heat of these agents using the equilibrium model by discarding sensor data, we would systematically overestimate how subcritical a network is due to the fact that the environment is interacting with the agent and vice versa.

3.2 Convergence of Evolution

Populations of different initial states follow distinct evolutionary strategies and most are able to solve the standard foraging task when evolved with the GA. Within the range of $\delta_{\text{init}} \in [-1, 1]$ as in the previous article (Prosi et al., 2021), all populations converge to a good fitness, but for $\delta_{\text{init}} \ll -1$ or $\delta_{\text{init}} \gg 1$ the GA sometimes cannot find suitable solutions. We observe evolution for 4,000 generations for populations initiated between the ranges of subcritical ($\beta_{\text{init}} \approx 32$, $\delta \approx -1.5$), critical ($\beta_{\text{init}} = 1$, $\delta \approx 0$), and supercritical ($\beta_{\text{init}} \approx 0.03$, $\delta \approx 1.5$) regimes. Critical populations begin to rapidly gain fitness from the first generation in every independent simulation run. (See Figure 4(a).) The gradual and stable increase of fitness of the initially critical population suggests that successful hill climbing on the fitness landscape is taking place. In contrast, for subcritical populations, fitness mainly evolves via random jumps and only about half of the simulations reach the same fitness as the critical populations after 4,000 generations (Figure 4(a); video: https://vimeo.com/547613948). Such fitness dynamics indicate a random search strategy which often leads to a population getting trapped in a local maxima for extended periods of time. Confirming the previous observations by Khajehabdollahi and Witkowski (2020), we see that



(a) Simple task: 12 neurons

(b) Simple task: 28 neurons

(c) Hard task: 12 neurons

Figure 4. Critically initialized populations can be successfully evolved in different circumstances, whereas for subcritically initialized populations, a harder task or an increased system size can significantly disrupt evolutionary dynamics. For each panel 10 initially critical ($\beta = 1$, green, top row) or initially subcritical ($\beta = 0.1$, blue, bottom row) populations evolve for 4,000 generations. The dashed line at fitness = 2 in the subcritical panels corresponds to the organisms' initial energy. It can be seen in the hard task with 12 neurons (bottom right) that the network is unable to achieve a fitness above its starting value.

supercritical populations, after an initial random period follow the same path as the critical ones, though highly supercritical models with $\delta > 1$ can sometimes struggle to find solutions.

For successfully evolvable populations, moderate changes in the complexity of the control network should not destroy the ability of the evolutionary algorithm to reach a good fitness. We test the differences in evolvability for initially critical and subcritical populations by changing the size of the network from 12 to 28 neurons. As in smaller networks, the initially critical populations rapidly evolve for all initial conditions. By generation 4,000 they even reach a slightly higher fitness than populations controlled by smaller networks and evolved for the same number of generations (Figure 4(b)). However, initially subcritical populations do not reach even half of their original fitness. We observe the same difference between the dynamical regimes when we increase the task's complexity requiring organisms to slow down to almost zero velocity in order to consume food particles (Figure 4(c), video: https://vimeo.com/547615705). In this harder task, the evolved populations' maximal fitness is expected to be lower than for the simple task. For the initially critical populations, we still observe the same hill-climbing dynamics. However, the initially subcritical populations stay at an energy level of precisely two. This signifies that they do not use the originally supplied energy for moving and remain static throughout all 4,000 generations, trapped in a local optimum.

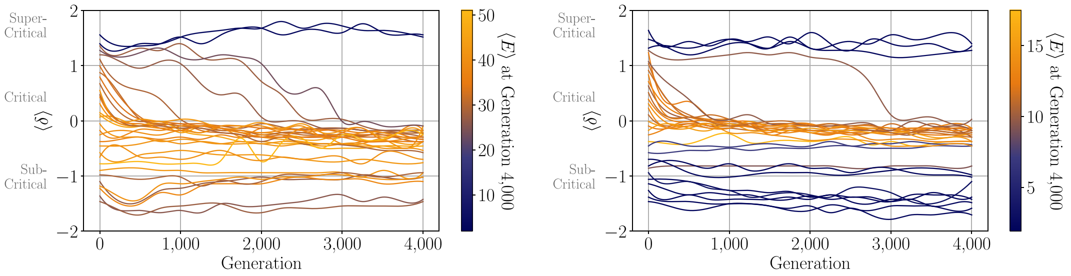
Overall, we see that although in simple tasks all populations can converge to approximately the same fitness, there exists a significant difference between the initially subcritical and initially critical/supercritical populations. Specifically, the convergence of evolution for critical populations is stable (all populations follow very similar fitness growth) and behave similarly regardless of network size or task complexity. For subcritical populations, the evolutionary dynamics resemble random search, which fails to find solutions in high-dimensional cases or for more complex tasks.

3.3 Evolution of the Dynamical Regime

Next, we investigate how the dynamical state of the populations changes during evolution. To do so, we select a wide range of initial dynamical regimes ($\delta \in [-1.5, 1.5]$) and examine how the dynamics of populations initialized in each of these regimes change throughout evolution via the GA. Regardless of their initial dynamics, almost all populations that manage to find solutions converge to the subcritical regime, albeit with different distances from the critical point (Figure 5). The populations that did not follow this convergence pattern were also ones that never discovered solutions to the task. We also observe that strongly subcritical populations ($\delta < -1$) and strongly supercritical populations ($\delta > 1$) generally achieve lower fitness in the simple task and are unable to solve the hard task.

In a basin spanning the near-critical regime, from moderately subcritical to moderately supercritical, populations rapidly change their dynamical regime and by generation 4,000 reach an intermediately subcritical state, whose δ we refer to as $\delta^* \approx -0.41$. This is a relatively broad dynamical regime whose evolutionary dynamics have different characteristics than the dynamics of deeply subcritical networks. Deeply subcritical populations with $\delta \ll \delta^*$ remain at their initial regimes for the GA, demonstrating a lack of evolutionary mobility and consequently are more likely to obtain lower fitnesses, whereas subcritical populations initialized at higher $0 \geq \delta \geq \delta^*$ can still approach δ^* which is correlated with the ability to solve the underlying task with high fitness. Similarly, deeply supercritical populations also struggled to change their dynamical regime; however, the supercritical populations that were able to optimize all converged to δ^*, much like the near-critical populations (Figure 5).

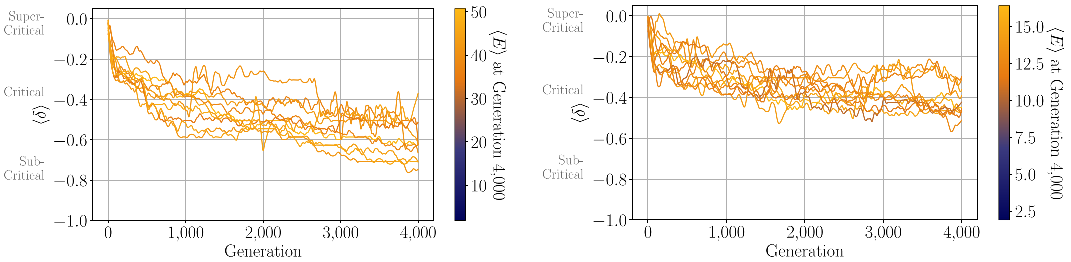
Task complexity determines the dynamical regime where evolution converges to. Specifically, when trying to solve the hard task, the agents converge to a smaller distance from criticality than when solving the original task. We check the evolution of the dynamical regime in both simple and hard tasks (Figure 6). We utilize the observation that almost all populations with an initial regime $\delta > \delta^*$ converge to similar values. Thus, we consider only initially critical populations. We obtain the distribution of dynamical states by considering 10 independent runs of evolution in both tasks after 4,000 generations. We take the mean of the top 30 most fit agents in each simulation, and



(a) Simple task

(b) Hard task

Figure 5. Genetic algorithm (GA): Changes in the distance to criticality over the course of evolution. 32 populations initiated at various distances to criticality (δ between −1.5 and 1.5) and evolved on a (a) simple task and (b) hard task. The color indicates their fitness at generation 4,000. Most populations with $\delta < 0$ remain at fitness 2 for the hard task, as well as a few simulations with $\delta > 1$, signifying no evolutionary progress.



(a) Simple task

(b) Hard task

Figure 6. Genetic algorithm (GA): State dynamics of 10 populations initialized at $\delta = 0$ and evolved on (a) a simple task and on (b) a hard task. All population solve the corresponding task but the populations trained on the hard task evolve to have δ values closer to criticality.

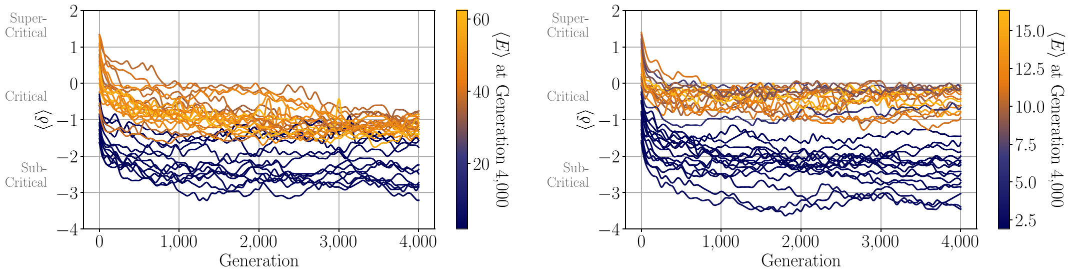
perform a Mann-Whitney U test to confirm that the δ values for the hard task are larger than the simple task ($p < 10^{-5}$), i.e., closer to the critical value. Specifically, the harder task results seem to consistently maintain a smaller distance to the critical point throughout evolution (Figure 9(a)).

Therefore, initiating an agent close to the critical regime is important when task complexity is unknown. We observe that the dynamical regime never changes towards supercriticality, but the subcritical convergence point can be at different distances from criticality. Thus, only starting near the critical point guarantees that the optimal dynamical state can be reached by evolution.

To verify that our results are not contingent on the specific implementation of the GA, we run the same experiment using a different optimization method—an evolution strategy, as described in Section 2.2.2. We re-run the experiments using the ES and obtain qualitatively similar results. In Figure 7, we observe that simulations near or above the critical point are able to discover high-scoring solutions in both the simple and the hard task (also see Figure 8). Furthermore, we once again observe that when initialized below a certain point, populations are unlikely to discover a good solution using the ES (even more than we observed for the GA).

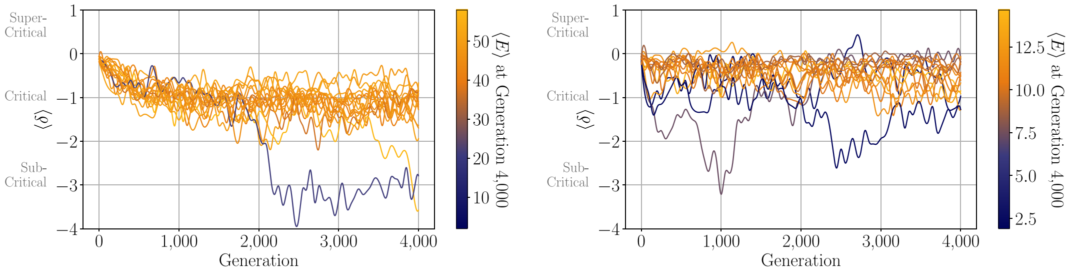
We observe that solutions to the simpler task converge to a more subcritical regime than for the hard task. However, the ES results in larger deviations from the critical point in the converged populations than the GA. This can be potentially attributed to the faster convergence of the dynamical state allowed by the ES.

We verify that the difference in the distance to criticality between simple and hard tasks is significant under both evolutionary algorithms, as shown in Figure 9(b). For each of the tasks, 16 independent populations of initially critical agents are evolved and the distribution of their δ values is presented. We compare the final δ values after 4,000 generations and confirm ($p < 10^{-2}$; Mann-Whitney U test) that the simpler task converges to a more subcritical regime than the hard



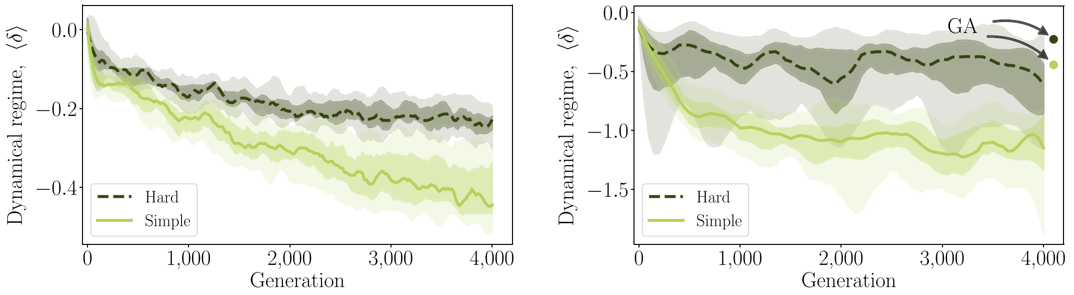
(a) Simple task (b) Hard task

Figure 7. Evolution strategy (ES): Changes in the distance to criticality over the course of evolution. 32 populations initiated at various distances to criticality (δ between −1.5 and 1.5) and evolved on (a) a simple task and on (b) a hard task. The colour indicates their fitness at generation 4,000. Similar, but more dramatic than the results from the GA, populations initialized with $\delta < 0$ suffer greatly in their ability to discover optimal solutions in both tasks. The ES can also be observed to have an overall tendency to become subcritical even when lacking selection pressure.



(a) Simple task (b) Hard task

Figure 8. Evolution strategy (ES): 16 populations initialized at $\delta = 0$ and evolved on (a) a simple task and on (b) a hard task. The populations trained on the hard task evolve to have δ values closer to criticality, similar to the results from the GA, albeit at larger values of δ.



(a) GA (b) ES

Figure 9. The dynamical regime of the initially critical population remains closer to the critical regime ($\delta = 0$) in the harder task than in the simple task throughout evolution, for both algorithms (tested for the final generation 4,000 with a Mann-Whitney U test, $p = 2.2 \times 10^{-6}$ and $p = 4.4 \times 10^{-3}$, for the GA and ES, respectively). 10 and 16 populations for the GA (a) and ES (b), respectively, are initiated in the critical regime and evolve for 4,000 generations. The 15th and 85th percentiles are lightly shaded, and the 33rd and 67th percentiles are more heavily shaded.

task for the ES as well. These results indicate that our findings are independent of the evolutionary algorithm used to solve the task.

3.4 Comparison of Evolutionary Algorithms

To understand how these two different families of evolutionary algorithms function differently, we compare their abilities in solving n-dimensional benchmark optimization problems (Rastrigin function (Equation 6), Rosenbrock function (Equation 7), and Sphere function (Equation 8)). The Rastrigin and Rosenbrock functions are difficult problems because of the existence of multiple local minima in the vicinity of the global minimum, whereas the sphere function has a unique minimum at 0, with smooth gradients towards it. The Rosenbrock function has relatively smooth gradients towards the broad local minima region, but it can be difficult to find the global minimum among them.

$$f_{\text{Rastrigin}}(\mathbf{x}) = An + \sum_{i=1}^{n} \left[x_i^2 - A\cos(2\pi x_i) \right] \tag{6}$$

$$f_{\text{Rosenbrock}}(\mathbf{x}) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2) + (1 - x_i)^2 \right] \tag{7}$$

$$f_{\text{Sphere}}(\mathbf{x}) = \sum_{i=1}^{n} x_i^2 \tag{8}$$

To avoid any inherent biases either algorithm might have towards solutions of a particular distribution, we translate the loss function (Equations 6 to 8) in space by a random Gaussian vector. To this end we can write the benchmark functions as a function of a new variable **z**, where $z_i = x_i + c_i$ where c_i are constants sampled from a $\mathcal{N}(0, 1)$ normal distribution, which results in randomization of the optimum's location.

Figure 10 compares the performance of the two algorithms. Due to the ES's approximation of the gradient, it is generally able to find the minimum of a smooth loss function faster than the GA (e.g., for the Sphere function, see Figure 10). However, owing to a fixed variance sampling, it can get stuck in a local solution as shown in a the Rastrigin function, where the GA surpasses the ES. Also for the Rosenbrock function, both algorithms only converge to the local minima of $\approx 10^{-3}$. For both the Rosenbrock and Sphere, the GA is orders of magnitude slower in finding comparable solutions.

Taking into account the significant differences of these two evolutionary algorithms, we strengthen the evidence for the universal utility of criticality for problem solving.

3.5 Generalizability

For successful biological systems robustness against environmental change is the paramount feature; therefore, it can be used to determine the success of evolved artificial organisms. We propose a simple measure to investigate how the model behaves outside of its explicit training conditions. Specifically, for a population trained with organism's lifespan parameter set at t_{train} we define

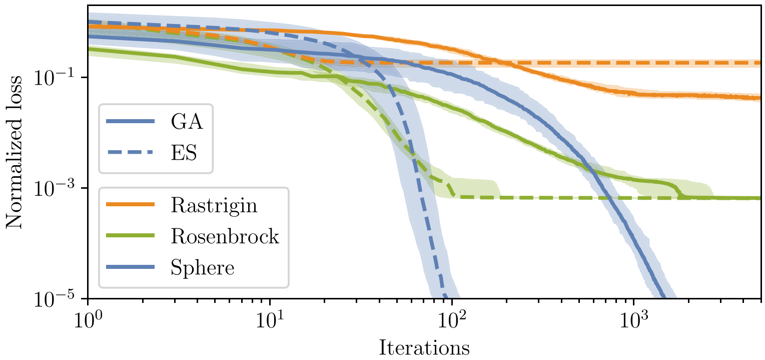


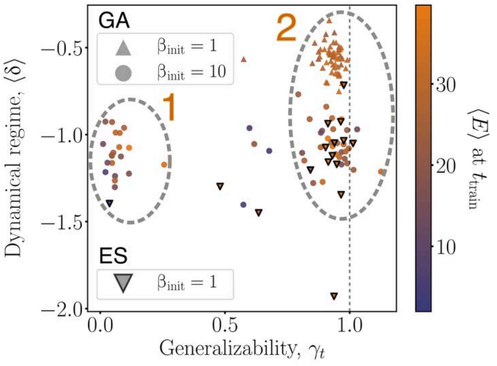
Figure 10. Comparisons of the loss dynamics of the ES and the GA on the Rastrigin, Rosenbrock, and Sphere functions for $n = 50$. The ES can take advantage of functions that have smooth gradients leading to its optima (Sphere, Rosenbrock), whereas the GA can overtake the ES for functions with multiple local optima (Rastrigin). Both algorithms failed to discover the global optima of the Rosenbrock function, and instead found the easier to find local minimum. The loss is normalized to start at 1 by dividing by the medians across independent runs, of the maximum value obtained throughout its evolution. The 25th to the 75th percentiles across 25 independent runs are shaded.

generalizability as the speed of growth of the average fitness if the organism's lifespan is extended to t_{extend}. Formally:

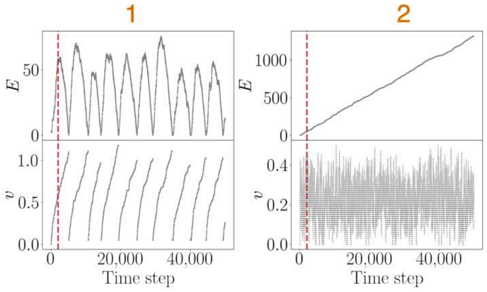
$$\gamma_t = \frac{\langle E_{t_{\text{train}}} \rangle / t_{\text{train}}}{\langle E_{t_{\text{extend}}} \rangle / t_{\text{extend}}}. \tag{9}$$

The *stable generalizability*, $\gamma_t = 1$ corresponds to linear growth whereas sublinear behaviour $\gamma_t \ll 1$ indicates possible overfitting to the particular organism's lifespan t_{train}.

We consider initially critical ($\delta \approx 0$, $\beta_{\text{init}} = 1$) and initially subcritical ($\delta \approx -1$, $\beta_{\text{init}} = 10$) populations evolved for 4,000 generations and then test their performance for an extended lifespan of $t_{\text{extend}} = 50{,}000$ time steps (instead of the $t_{\text{train}} = 2{,}000$). As reported in previous sections, the critical populations evolved with the GA converge to $\delta \approx -0.41$, and they all have a similar fitness after training. Interestingly, when increasing the organisms' lifespan, the fitness of the critical population continues to grow linearly, signifying almost perfect generalizability. About half of the subcritical populations reach the same fitness level. However, the subcritical populations split up into two clusters: cluster one with generalizability close to 0, and cluster two with generalizability close to 1 (Figure 11(a)). Surprisingly, there is no difference in fitness between these two clusters. We also evolve the same populations with the ES algorithm and obtain a similar picture for the critical population: Almost all populations attain generalizability close to 1. The initially subcritical



(a)



(b)

Figure 11. Populations initialized at criticality find solutions that generalize beyond their training condition, whereas subcritical populations often overfit and fail when the organism's lifespan is changed. (a) 54 populations of each type (triangles: initially critical; circles: initially subcritical). After 4,000 generations of the GA, all critical populations reach a high fitness (indicated by colour) and nearly perfect (with one exception) generalizability γ_t (Equation 9). The initially subcritical organisms split between badly generalizing cluster 1 (19 populations), generalizable cluster 2 (28 populations), and 7 populations not assigned to any cluster. This split is not predicted by their attained fitness. For the ES (indicated by the triangles pointing down with black border) the variability of initially critical solutions is larger (13 out of 16 are in cluster 2, one in cluster 1, and two in-between), and initially subcritical organisms $\beta_{\text{init}} = 10$ do not evolve to any reasonable fitness. (b) Energy and velocity as a function of time for representative examples of the organisms from cluster 1 and clusters 2 (marked in panel (a)). The dashed orange line denotes the training lifespan $t_{\text{train}} = 2{,}000$.

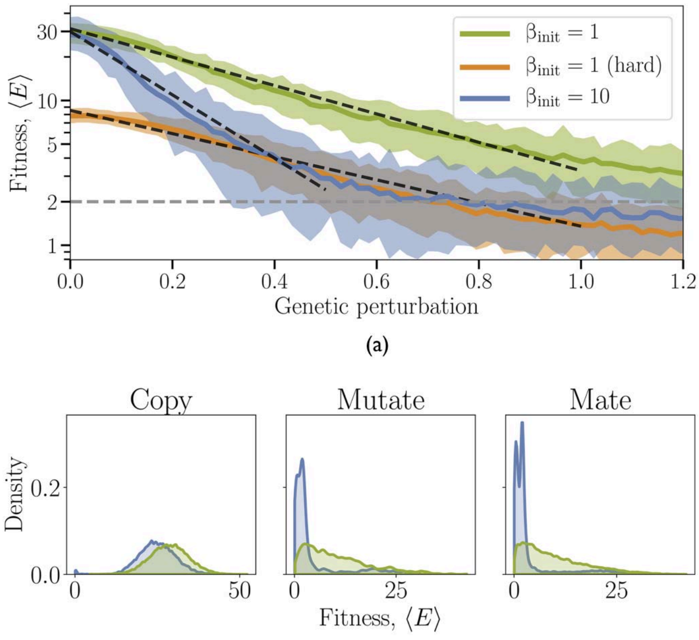
populations ($\beta_{\text{init}} = 10$), however, fail to evolve to a compatible fitness, and therefore we exclude them from the figure. Out of 16 populations, one did not generalize at all (in cluster 1) but it also did not reach a similar fitness level ($\langle E \rangle_{t_{\text{train}}} \approx 20$), and two reach an intermediate state. Notably, the 3 populations that did not generalize for the ES were some of the most subcritical solutions found.

To understand the difference we look more precisely at both clusters (Figure 11(b)). Organisms in cluster 1 reach their maximal fitness/average lifetime energy (sometimes higher than in cluster 2) at the end of their lives, but often quickly lose fitness when tested beyond their training environment. The velocity profile v offers an explanation for the bad generalization. The organisms from cluster 1 follow the strategy to increase the velocity permanently until the end of their training lifespan (Figure 11(b)). However, moving with such a high velocity is not compatible with the energy influx from feeding, and they break down shortly after the end of their training lifespan; this demonstrates that these organisms overfit the training conditions. The generalizable populations (cluster 2) have a much more complex velocity profile that accelerates and decelerates often, in contrast to its fitness, which grows consistently and linearly beyond the lifespan of their training environment.

Overall, the initialization in the critical regime results in almost perfect generalizability of evolved populations, whereas initially strongly subcritical populations risk overfit of their training conditions.

3.6 Effect of Genetic Perturbations on the Fitness

Next, we examine the stability of the evolved organisms to genetic perturbations. We apply genetic perturbations of different magnitudes to the evolved organisms of initially critical $\beta_{\text{init}} = 1$ and subcritical β_{init} populations. We perturb all weights of the connectivity matrix by randomly adding or subtracting a number f_{pert} and then evaluate the fitness of the resulting organism. As expected, we find that fitness decays rapidly with perturbation magnitude for both batches of populations; however, the subcritical ones decay much faster (Figure 12(a)). By characterizing this fitness decay,



(a)

(b)

Figure 12. Initially critical populations show larger genotypic stability than initially subcritical ones. (a) The phenotype (fitness) as a function of the genotypic perturbation (changes in connectivity) for two initial conditions: $\beta_{\text{init}} = 1$ and $\beta_{\text{init}} = 10$. For the initially critical model we also plot the change in fitness for the hard task (orange line). Dashed black lines indicate exponential fit with exponent −2.26 (critical), −1.84 (critical, hard), and −5.03 (subcritical). (b) The histograms of the fitness values for nearly fully evolved agents (between generation 3,500 and 4,000), categorized according to the last evolutionary operator (copy, mutate, or mate) that was applied to them. The $\beta_{\text{init}} = 10$ agents are less likely to remain fit when their genotype is changed by mutation or mating.

we can understand better the smoothness of the genotype–phenotype map of these optimized solutions and how different solutions can be found with varying degrees of robustness. We evaluate the fitness decay by the slope of an exponential function fitted to the fitness.

For the hard task, only the fitness decay of the initially critical populations can be evaluated ($\alpha = -1.84$), as the subcritical population was unable to solve this task. For the simple task, $\alpha = -2.26$ and $\alpha = -5.03$ for the initially critical and subcritical populations, respectively. The subcritical population decays in fitness at more than double the rate than the critical population, indicating a much higher sensitivity of subcritical systems to perturbations. Interestingly, the fitness decay of both critical populations in the easy task and hard task are similar.

The evolutionary algorithm is a source of constant genetic perturbations that are necessary in the beginning of evolution but can become detrimental later. We consider the individual effect of the evolutionary operators (copy, mutate, and mate) on the resulting fitness of the organisms. The variability of fitness for copying simply reflects the natural variability in community fitness rankings and organism behaviour. However, both mating and mutation in fully evolved subcritical populations typically results in a fitness close to 2—signifying totally unfit organisms (Figure 12(b)). At the same time, initially critical organisms retain diverse fitness values after mutation and mating, some being close to the optimum. This phenotypic diversity allows the originally critical populations to retain their evolvability as opposed to the rigid search performed by strongly subcritical populations which have more discontinuous genotype–phenotype landscapes.

4 Discussion

We demonstrate that in various scenarios, evolving populations of agents converge to a moderately subcritical state with the resulting deviation from criticality depending on the task's difficulty. This might appear to be a contradiction to the previous studies, suggesting that operating close to criticality is optimal for natural systems (Mora & Bialek, 2011; Muñoz, 2018; Roli et al., 2018). However, a recent body of research showed that for simple tasks, operating at some distance to criticality might be an optimal solution for the sensitivity/stability tradeoff (Cramer et al., 2020; Hidalgo et al., 2014; Tomen et al., 2014; Villegas et al., 2016).

We observe that the distance from criticality affects an agent's ability to solve complex tasks and to robustly evolve generalizable behaviour, validated using two different evolutionary approaches: a genetic algorithm and an evolutionary strategy. Specifically, we observe that *slightly subcritical* populations are evolvable for different complexities of the control network and task, whereas *strongly subcritical* populations fail in both algorithms. Interestingly, the ES is more successful in optimizing strongly supercritical populations but less so for strongly subcritical ones. Given that solving complex tasks and being adaptive are crucial in natural environments, we propose that living systems operate in the subcritical regime in close proximity to the critical point. Moreover, we show that the optimal regime moves closer to criticality as we increase the task difficulty, which suggests that the optimal distance from criticality varies. These findings are confirmed by Cramer et al. (2020) as well as Villegas et al. (2016), who showed that the optimal distance from criticality in the subcritical regime decreases for higher task complexity or larger system size.

We further observe that populations can only become more subcritical during evolution corresponding to a more ordered phase. They fail to become more critical (more disordered) even when this would have eventually led to superior behaviour, a phenomenon that we suspected to be an artefact of the GA used in Prosi et al. (2021). However, this does not seem to be the case as confirmed here by using the ES, which replicates this phenomenon. As it is a priori unknown which distance from criticality will be optimal when evolving for a new task, initializing at the critical point could be the only way for the evolutionary process to descend to the optimal regime. However, in the long run this would require some sub-populations to always maintain closeness to criticality. How this can be achieved for neuronal networks is a subject of vivid research (for a review, see Buendía et al., 2020; Kinouchi et al., 2020; Zeraati et al., 2021), and for the embodied Ising agents it remains open for further investigation. Maintaining evolvability in simpler systems was, for instance, achieved by switching between different rough energy landscapes (Wang & Dai, 2019). The inhomogeneity of the environment and coevolution can also contribute to the preservation of the critical regime (Hidalgo et al., 2014). Overall, the maintenance of evolvability throughout evolution is an important question beyond the embodied Ising agents studied here.

Our results extend and partly revise the earlier findings of Khajehabdollahi and Witkowski (2020), which reported a superior evolvability of critical populations and an approximate convergence to criticality during evolution. We confirm that the critical regime allows reliable evolvability, and additionally, we extend our understanding by considering a set of tasks and architectures in the model. However, our more precise procedure to infer the dynamical regime and the fine sampling of initial conditions uncovered additional complex dynamics. Namely, critical populations of Ising agents converge to the subcritical regime, and the distance to criticality depends on the task complexity. We extend our earlier work (Prosi et al., 2021) and verify that the results are not contingent on the specifics of our GA by comparing them to results generated by an ES, an instance of a different family of evolutionary algorithms. We confirm that our major findings hold under both optimization algorithms. The ES, however, shows stronger sensitivity to its hyper-parameters, but optimizes the fitness faster. A future study could potentially compare these results with an algorithm with an adaptive σ such as parameter-exploring policy gradients (Sehnke et al., 2010) or covariance matrix adaptation evolution strategy.

We also propose a new way to investigate capabilities of the resulting organisms by defining generalizability and genetic stability measures. Both measures reveal the benefits of staying close to the critical state beyond a simple fitness comparison.

Acknowledgments

This work was supported by a Sofja Kovalevskaja Award from the Alexander von Humboldt Foundation. SK and EG thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for support. We acknowledge the support from the BMBF through the Tübingen AI Center (FKZ: 01IS18039A).

Author Contributions

Jan Prosi and Sina Khajehabdollahi contributed equally to this research.

References

Aguilera, M., & Bedia, M. G. (2017). Criticality as it could be: Organizational invariance as self-organized criticality in embodied agents. *ECAL 2017: Proceedings of fourteenth European conference on artificial life* (pp. 21–28). MIT Press. https://doi.org/10.7551/ecal_a_009

Aldana, M., Balleza, E., Kauffman, S., & Resendiz, O. (2007). Robustness and evolvability in genetic regulatory networks. *Journal of Theoretical Biology*, *245*(3), 433–448. https://doi.org/10.1016/j.jtbi.2006.10.027

Balleza, E., Alvarez-Buylla, E. R., Chaos, A., Kauffman, S., Shmulevich, I., & Aldana, M. (2008). Critical dynamics in genetic regulatory networks: Examples from four kingdoms. *PLOS ONE*, *3*(6), Article e2456. https://doi.org/10.1371/journal.pone.0002456, PubMed: 18560561

Beggs, J. M. (2007). The criticality hypothesis: How local cortical networks might optimize information processing. *Philosophical Transactions of the Royal Society A*, 366(1864), 329–343. https://doi.org/10.1098/rsta.2007.2092, PubMed: 17673410

Beggs, J. M., & Plenz, D. (2004). Neuronal avalanches are diverse and precise activity patterns that are stable for many hours in cortical slice cultures. *Journal of Neuroscience*, *24*(22), 5216–5229. https://doi.org/10.1523/JNEUROSCI.0540-04.2004

Bertschinger, N., & Natschläger, T. (2004). Real-time computation at the edge of chaos in recurrent neural networks. *Neural Computation*, *16*(7), 1413–1436. https://doi.org/10.1162/089976604323057443, PubMed: 15165396

Boedecker, J., Obst, O., Lizier, J. T., Mayer, N. M., & Asada, M. (2012). Information processing in echo state networks at the edge of chaos. *Theory in Biosciences*, *131*(3), 205–213. https://doi.org/10.1007/s12064-011-0146-8

Buendía, V., di Santo, S., Bonachela, J. A., & Muñoz, M. A. (2020). Feedback mechanisms for self-organization to the edge of a phase transition. *Frontiers in Physics*, *8*, Article 333. https://doi.org/10.3389/fphy.2020.00333

Cavagna, A., Cimarelli, A., Giardina, I., Parisi, G., Santagati, R., Stefanini, F., & Viale, M. (2010). Scale-free correlations in starling flocks. *Proceedings of the National Academy of Sciences*, *107*(26), 11865–11870. https://doi.org/10.1073/pnas.1005766107, PubMed: 20547832

Chaté, H., & Muñoz, M. A. (2014). Insect swarms go critical. *Physics*, *7*, Article 120.

Cramer, B., Stöckel, D., Kreft, M., Wibral, M., Schemmel, J., Meier, K., & Priesemann, V. (2020). Control of criticality and computation in spiking neuromorphic networks with plasticity. *Nature Communications*, *11*, Article 2853. https://doi.org/10.1038/s41467-020-16548-3, PubMed: 32503982

De Jong, K. A. (2006). *Evolutionary computation: A unified approach.* MIT Press. https://doi.org/10.1145/1274000.1274109

De Palo, G., Yi, D., & Endres, R. G. (2017). A critical-like collective state leads to long-range cell communication in dictyostelium discoideum aggregation. *PLOS Biology*, *15*(4), Article e1002602. https://doi.org/10.1371/journal.pbio.1002602, PubMed: 28422986

Halley, J. D., Burden, F. R., & Winkler, D. A. (2009). Stem cell decision making and critical-like exploratory networks. *Stem Cell Research*, *2*(3), 165–177. https://doi.org/10.1016/j.scr.2009.03.001, PubMed: 19393588

Hidalgo, J., Grilli, J., Suweis, S., Munoz, M. A., Banavar, J. R., & Maritan, A. (2014). Information-based fitness and the emergence of criticality in living systems. *Proceedings of the National Academy of Sciences*, *111*(28), 10095–10100. https://doi.org/10.1073/pnas.1319166111, PubMed: 24982145

Kauffman, S. A. (1993). *The origins of order: Self-organization and selection in evolution.* Oxford University Press. https://doi.org/10.1007/978-94-015-8054-0_8

Kauffman, S. [A.], & Levin, S. (1987). Towards a general theory of adaptive walks on rugged landscapes. *Journal of Theoretical Biology*, *128*(1), 11–45. https://doi.org/10.1016/S0022-5193(87)80029-2

Khajehabdollahi, S., & Witkowski, O. (2020). Evolution towards criticality in Ising neural agents. *Artificial Life*, *26*(1), 112–129. https://doi.org/10.1162/artl_a_00309, PubMed: 32027529

Kinouchi, O., & Copelli., M. (2006). Optimal dynamical range of excitable networks at criticality. *Nature Physics*, *2*(5), 348–352. https://doi.org/10.1038/nphys289

Kinouchi, O., Pazzini, R., & Copelli, M. (2020). Mechanisms of self-organized quasicriticality in neuronal network models. *Frontiers in Physics*, *8*, Article 530. https://doi.org/10.3389/fphy.2020.583213

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, *21*(6), 1087–1092. https://doi.org/10.1063/1.1699114

Mora, T., & Bialek, W. (2011). Are biological systems poised at criticality? *Journal of Statistical Physics*, *144*(2), 268–302. https://doi.org/10.1007/s10955-011-0229-4

Muñoz, M. A. (2018). Colloquium: Criticality and dynamical scaling in living systems. *Reviews of Modern Physics*, *90*(3), Article 031001. https://doi.org/10.1103/RevModPhys.90.031001

Najarro, E., & Risi, S. (2020). Meta-learning through Hebbian plasticity in random networks. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), *NIPS '20: Proceedings of the 34th conference on neural information processing systems. Advances in neural information processing systems 33* (pp. 20719–20731). https://proceedings.neurips.cc/paper/2020/file/ee23e7ad9b473ad072d57aaa9b2a5222-Paper.pdf

Poel, W., Daniels, B. C., Sosna, M. M. G., Twomey, C. R., Leblanc, S. P., Couzin, I. D., & Romanczuk, P. (2022). Subcritical escape waves in schooling fish. *Science Advances*, 8(25), Article eabm6385. https://doi.org/10.1126/sciadv.abm6385

Priesemann, V., Wibral, M., Valderrama, M., Pröpper, R., Le Van Quyen, M., Geisel, T., Triesch, J., Nikolić, D., & Munk, M. H. J. (2014). Spike avalanches in vivo suggest a driven, slightly subcritical brain state. *Frontiers in Systems Neuroscience*, *8*, Article 108. https://doi.org/10.3389/fnsys.2014.00108, PubMed: 25009473

Prosi, J., Khajehabdollahi, S., Giannakakis, E., Martius, G., & Levina, A. (2021). The dynamical regime and its importance for evolvability, task performance and generalization. *ALIFE 2021: Proceedings of the 2021 conference on artificial life* (Article 79). https://doi.org/10.1162/isal_a_00412

Rämö, P., Kauffman, S., Kesseli, J., & Yli-Harja, O. (2007). Measures for information propagation in boolean networks. *Physica D: Nonlinear Phenomena*, *227*(1), 100–104. https://doi.org/10.1016/j.physd.2006.12.005

Rämö, P., Kesseli, J., & Yli-Harja, O. (2006). Perturbation avalanches and criticality in gene regulatory networks. *Journal of Theoretical Biology*, *242*(1), 164–170. https://doi.org/10.1016/j.jtbi.2006.02.011, PubMed: 16574157

Roli, A., Villani, M., Filisetti, A., & Serra, R. (2018). Dynamical criticality: Overview and open questions. *Journal of Systems Science and Complexity*, *31*(3), 647–663. https://doi.org/10.1007/s11424-017-6117-5

Salimans, T., Ho, J., Chen, X., Sidor, S., & Sutskever, I. (2017). Evolution strategies as a scalable alternative to reinforcement learning. ArXiv, https://doi.org/10.48550/arXiv.1703.03864

Schneidman, E., Berry, M. J., Segev, R., & Bialek, W. (2006). Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, *440*(7087), 1007–1012. https://doi.org/10.1038/nature04701

Sehnke, F., Osendorfer, C., Rückstieß, T., Graves, A., Peters, J., & Schmidhuber, J. (2010). Parameter-exploring policy gradients. *Neural Networks*, *23*(4), 551–559. https://doi.org/https://doi.org/10.1016/j.neunet.2009.12.004, PubMed: 20061118

Tkačik, G., Mora, T., Marre, O., Amodei, D., Palmer, S. E., Berry, M. J., II, & Bialek, W. (2015). Thermodynamics and signatures of criticality in a network of neurons. *Proceedings of the National Academy of Sciences*, *112*(37), 11508–11513. https://doi.org/10.1073/pnas.1514188112, PubMed: 26330611

Tomen, N., Rotermund, D., & Ernst, U. (2014). Marginally subcritical dynamics explain enhanced stimulus discriminability under attention. *Frontiers in Systems Neuroscience*, *8*, Article 151. https://doi.org/10.3389/fnsys.2014.00151, PubMed: 25202240

Villegas, P., Ruiz-Franco, J., Hidalgo, J., & Muñoz, M. A. (2016). Intrinsic noise and deviations from criticality in Boolean gene-regulatory networks. *Scientific Reports*, *6*, Article 34743. https://doi.org/10.1038/srep34743

Wang, S., & Dai, L. (2019). Evolving generalists in switching rugged landscapes. *PLOS Computational Biology*, *15*(10), Article e1007320. https://doi.org/10.1371/journal.pcbi.1007320, PubMed: 31574088

Wierstra, D., Schaul, T., Glasmachers, T., Sun, Y., Peters, J., & Schmidhuber, J. (2014). Natural evolution strategies. *The Journal of Machine Learning Research*, *15*(1), 949–980.

Wierstra, D., Schaul, T., Peters, J., & Schmidhuber, J. (2008). Natural evolution strategies. In *Proceedings of the 2008 Congress on Evolutionary Computation. CEC'08* (pp. 3381–3387). IEEE. https://doi.org/10.1109/CEC.2008.4631255

Wilting, J., & Priesemann, V. (2019). Between perfectly critical and fully irregular: A reverberating model captures and predicts cortical spike propagation. *Cerebral Cortex*, *29*(6), 2759–2770. https://doi.org/10.1093/cercor/bhz049

Zeraati, R., Priesemann, V., & Levina, A. (2021). Self-organization toward criticality by synaptic plasticity. *Frontiers in Physics*, *9*, Article 103. https://doi.org/10.3389/fphy.2021.619661

Zierenberg, J., Wilting, J., Priesemann, V., & Levina, A. (2020). Tailored ensembles of neural networks optimize sensitivity to stimulus statistics. *Physical Review Research*, *2*(1), Article 013115. https://doi.org/10.1103/PhysRevResearch.2.013115

Appendices

Appendix 1: Effects of Thermalization Time

The Ising networks have a time span during which the system can adapt to the new inputs—the thermalization time. In Figure 13 we analyze the dependency of the fitness of individuals evolved on this parameter. We find that the value of 20 yields optimal fitness; however, we have chosen the value of 10 for computational reasons as it provides a good compromise between computational performance and achieved fitness in a given number of generations. Future investigations can

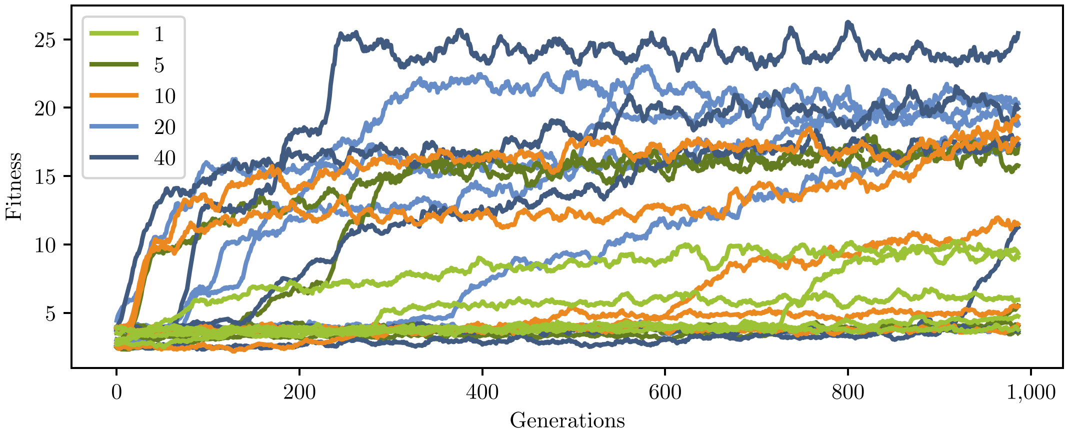


Figure 13. Effect of thermalization time on fitness. The effect of different thermalization times (1, 5, 10, 20, and 40 thermalization steps) is tested for 5 independent populations each, evolved using the GA in the simple task.

be directed to uncover how the thermalization time influences the optimal dynamic regime after convergence or before the beginning of the evolution.

Appendix 2: Distribution of Distances to Criticality

In section 3.3 we discuss the evolution of the dynamical regime as measured by the distance to criticality, summarized by the parameter δ. Specifically, we compare the final values of δ after 4,000 generations of evolution, for both the ES and the GA, for the simple task and the hard task. A box plot and histogram of the data from the final generations of these simulations are plotted here. For each independent simulation, the top 30 most fit agents are selected and their δ values averaged. We have 54 simulations using the GA and 16 using the ES. (In Figure 9, the lines are generated from a data set with 10 and 16 simulations for the GA and ES, respectively. Here we use the larger data set of 54 simulations for the GA, which only has δ calculated for its final generation.) A Mann-Whitney U test is used to check if the δ values of the hard task are higher than the simple task. For both the GA and ES, the test shows the hard task has larger δs than the simple task, with $p = 2.2 \times 10^{-6}$ and $p = 4.4 \times 10^{-3}$, respectively. See Figure 14 and Figure 15.

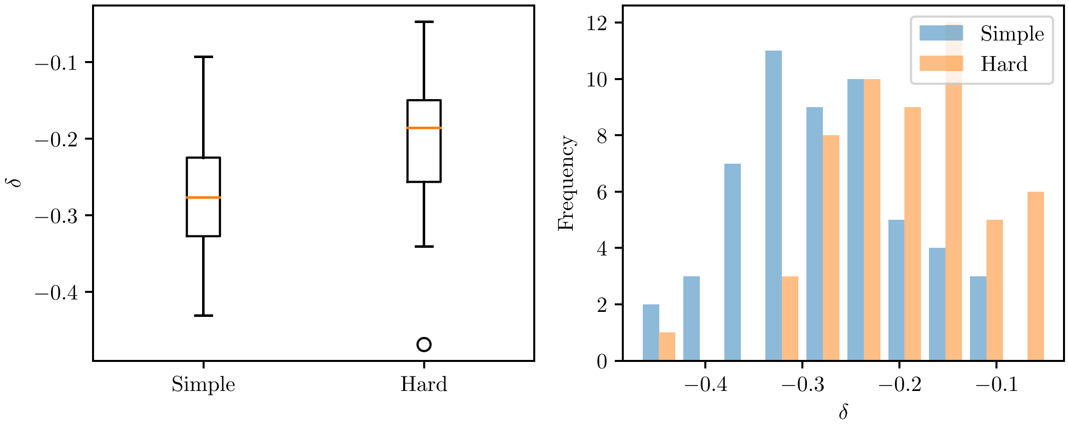


Figure 14. Distribution of the final δ values for 54 independent simulations using the GA. Each simulation has a population of 50, of which only the top 30 most fit individuals have their δs recorded and then averaged. The hard task has larger δ values than the simple task with $p = 2.2 \times 10^{-6}$ according to a Mann-Whitney U test.

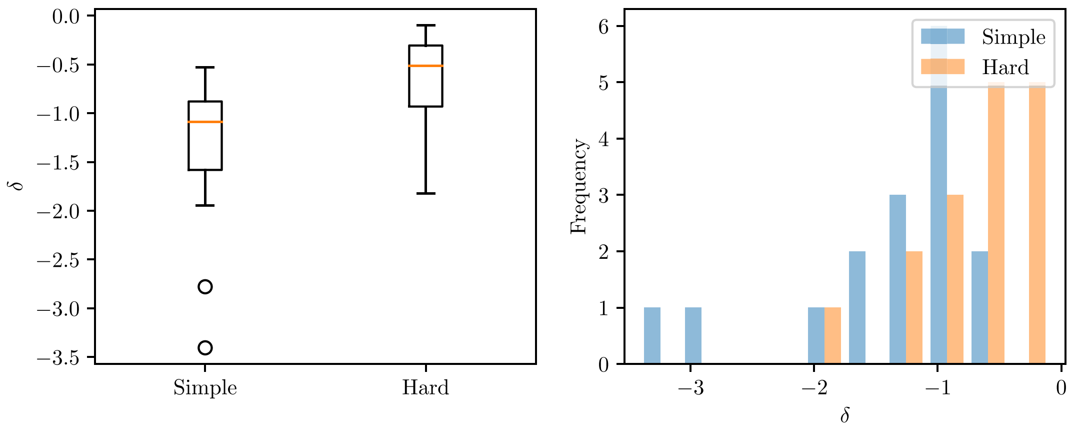


Figure 15. Distribution of the final δ values for 16 independent simulations using the ES. Each simulation has a population of 50, of which only the top 30 most fit individuals have their δs recorded and then averaged. The hard task has larger δ values than the simple task with $p = 4.4 \times 10^{-3}$ according to a Mann-Whitney U test.

A.3 Locally adaptive cellular automata for goal-oriented self-organization

Locally adaptive cellular automata for goal-oriented self-organization

Sina Khajehabdollahi[1], Emmanouil Giannakakis[1,2], Victor Buendía[1,2], Georg Martius[4], Anna Levina[1,2,3]

[1]Department of Computer Science, University of Tübingen, Tübingen, Germany
[2]Max Planck Institute for Biological Cybernetics, Tübingen, Germany
[3]Bernstein Center for Computational Neuroscience Tübingen, Tübingen, Germany
[4] Max Planck Institute for Intelligent Systems, Tübingen, Germany
sina.abdollahi@gmail.com

Abstract

The essential ingredient for studying the phenomena of emergence is the ability to generate and manipulate emergent systems that span large scales. Cellular automata are the model class particularly known for their effective scalability but are also typically constrained by fixed local rules. In this paper, we propose a new model class of adaptive cellular automata that allows for the generation of scalable and expressive models. We show how to implement computation-effective adaptation by coupling the update rule of the cellular automaton with itself and the system state in a localized way. To demonstrate the applications of this approach, we implement two different emergent models: a self-organizing Ising model and two types of plastic neural networks, a rate and spiking model. With the Ising model, we show how coupling local/global temperatures to local/global measurements can tune the model to stay in the vicinity of the critical temperature. With the neural models, we reproduce a classical balanced state in large recurrent neuronal networks with excitatory and inhibitory neurons and various plasticity mechanisms. Our study opens multiple directions for studying collective behavior and emergence.

Introduction

Cellular automata (CA) are simple models of computation where cells organized in a regular grid update their state according to rules that are local functions of the cell's neighborhood (von Neumann, 1966; Chopard and Droz, 1998). It has long been observed that CAs can exhibit complex pattern formation and highly non-trivial dynamics. The simplicity of the basic concept, combined with their extreme versatility make CAs a great tool for studying the phenomena of emergence and complexity in a mathematical framework (Wolfram, 1984). CAs have been used to model a variety of physical (Vichniac, 1984; Toffoli, 1984), biological (Langton, 1986; Turing, 1952; Mordvintsev et al., 2020; Farner et al., 2021), and more recently, differentiable, goal-oriented artificial intelligence phenomena (Chan, 2020; Mordvintsev et al., 2020; Variengien et al., 2021; Randazzo et al., 2020; Pontes-Filho et al., 2022).

The success of CAs in modeling complex physical and biological phenomena is perhaps in no small part due to the model's inherent inductive bias that reflects reality reasonably well. Any CA model is inherently local and computes and distributes information locally, a constraint that also exists strongly in the real world *(ignoring spooky action at a distance)*. Due to this constraint, CAs optimized for certain problems will always solve them via collective local interaction. CAs, therefore, allow us to study how self-organizing systems with only local interactions can give rise to global structure and complexity, a key characteristic of living matter.

However, for the very reason they are efficient, the expressive capabilities of CAs are constrained by their design principles. Namely, the homogeneity of the update rule in space and time forces every cell in the system to update according to the same rule. This means that CA models, which can be thought of as mesoscopic/coarse-grained models of phenomena, are fixed in the level of abstraction that they are modeling. Some work on multi-scale cellular automata has shown the capacity of such models to generate multi-scale patterns when multiple rules are interacting or competing (McCabe, 2010; Rampe, 2021; Slackermanz, 2021). These systems demonstrate how context-specific heterogeneity in the application of these CA rules can allow for another qualitative change in the emergent dynamics. This multi-scale property may, in fact, be crucial to the top-down organizational properties of biology (Pezzulo and Levin, 2016). Most existing models still only have a fixed set of update rules that are then applied selectively and cannot be flexibly modified as in nature. There are variations of CAs that address this homogeneity in different ways, such as probabilistic CAs (Louis and Nardi, 2018) or models of neural cellular automata manifolds (Hernandez et al., 2021). Here, we present a more general description of adaptive CAs that can be extended to a variety of domains.

Unfortunately, much of the CAs' utility comes from their computational efficiency and simplicity. Implementing heterogeneity or adaptation in CAs can make them too computationally expensive at the scales they require to become interesting. Here, we propose a method for designing adaptive cellular automata that minimizes the impact of adaptation on computational efficiency (Fig. 2) by taking advantage of advances in computational power. In particular, with modern GPU hardware and fast linear algebra libraries it becomes very cheap to write our CAs as highly parallelizable matrix operations such as convolutions. In turn, this allows for a massive increase in the scale, expressivity, interactivness and ability to train/differentiate such models to the extent that they can be run on personal computers with high frame rates.

The methods discussed in this paper are meant as building blocks for designing models of collective systems that exhibit emergence. We follow here the principle of building intuitions about such systems through the process of constructing them (von Neumann, 1966). This will then further allow us to study these systems in quantitative ways, build better models, and guide our theories. Having toy models of emergence that simulate or approximate the levels of emergence we observe in nature is a crucial step toward expanding our theories of complexity and self-organization. To that end, we implement as adaptive CAs three known foundational models of collective dynamics, the 2D Ising model, a Wilson-Cowan rate neural network, and a leaky integrate-and-fire spiking neural network. As these models can often be generalized across domains as generic models of interacting systems, our aim is to explore methods of local adaptation that can self-organize these general systems towards desirable macroscopic states.

Method

Update rules for cellular automata are traditionally fixed and applied identically and in parallel at every cell on the grid. By reusing a fixed rule that is always applied identically, CAs can take advantage of highly parallelizable algorithms with minimal memory costs. These update rules can often be written as sequences of matrix multiplication operations, which when implemented with modern linear algebra libraries and GPUs offer very efficiently computed models that scale well. In this framework, convolutions are a natural choice to accomplish the types of local calculations required by cellular automata (Gilpin, 2019; Mordvintsev et al., 2020). However, 2D convolutions (Procedure 1) use a fixed kernel, which means that a CA that makes use of these operations will also have a fixed update rule at all times, see Fig. 1a. However, with the relative growth of GPU memory capacity and computational power, it has become increasingly feasible to embed localized rule parameters in memory and add a new dimension of heterogeneity to these models (Hernandez et al., 2021).

To allow cellular automata to have *adaptive* update rules that can change flexibly as the system evolves in time, we define recursive update rules where the rule parameters are embedded into the state of the system, see Fig. 1b (Procedure 2). To do this, we can concatenate the state $\boldsymbol{\sigma}_i(t)$ of a cell with the parameters $\boldsymbol{\theta}_i(t)$ of a local update rule, such that the state and update rule of a cell at time t is fully defined by

$$\boldsymbol{s}_i(t) = [\boldsymbol{\sigma}_i(t), \boldsymbol{\theta}_i(t)]. \tag{1}$$

Whereas in a classical CA, the update rule $f(\boldsymbol{\sigma}_i(t))$ would be fixed, with adaptive CAs, the update rule

$$\boldsymbol{s}_i(t+1) = f(\boldsymbol{s}_i(t)) = f(\boldsymbol{\sigma}_i(t), \boldsymbol{\theta}_i(t)) \tag{2}$$

is local and parameterized by $\boldsymbol{\theta}_i(t)$. There are a number of different ways one can parameterize a local update rule, for example, the weights of the forces a cell feels from its neighbors, a decision tree of a set of locally context-dependent update rules, and so on. The toy models shown here are all implemented using the PyTorch library (Paszke et al., 2019).

Procedure 1 2D Convolution

```
function CONV2D($image, kernel$)
    $patches \leftarrow$ unfold($image, patchSize$)
    for $p_{ij}$ in $patches$ do
        $out_{ij} \leftarrow$ sum($p_{ij} \odot kernel$)
    end for
    return $out$
end function
```

Procedure 2 Generalized sliding-window function

```
function LOCAL($image, \theta$)
    $patches \leftarrow$ unfold($image, patchSize$)
    $\Theta \leftarrow$ unfold($image, patchSize$)
    for $p_{ij}, \theta_{ij}$ in zip($patches, \Theta$) do
        $out_{ij} \leftarrow$ adaptive_rule($\theta_{ij}$)($p_{ij}$)
    end for
    return $out$
end function
```

Ultimately, there is still a bedrock level in which the model is fixed that is determined by the nature of the update rule's own update rule. However, by making the update rule adaptive, we can allow CAs to navigate through a much larger space of dynamical systems, giving it the possibility to navigate its phase space according to some driving principle or goal.

Examples

Self-organized criticality in the Ising CA

Phase transitions are fundamental phenomena in the study of complex systems as they demonstrate how matter changes its internal structure under varying conditions. Understanding the boundaries of different phases of organization and the conditions under which a system transitions from one phase to another is central to the understanding of both physical and living systems. Furthermore, there is increasing evidence that it is precisely at these boundaries between

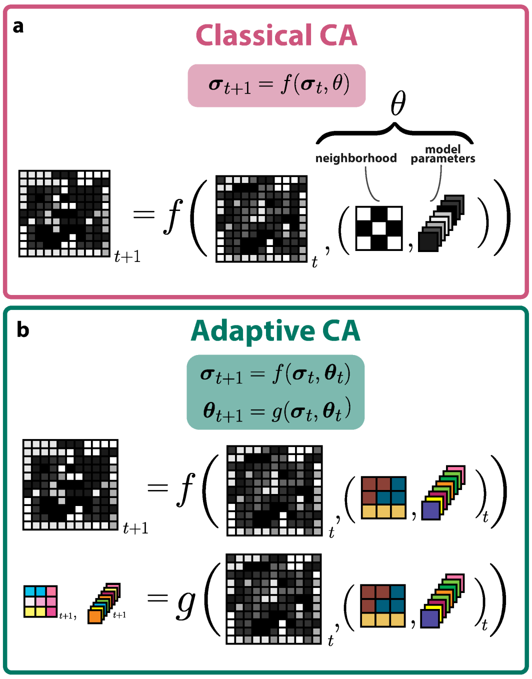

a
Classical CA
$\boldsymbol{\sigma}_{t+1} = f(\boldsymbol{\sigma}_t, \theta)$
θ
neighborhood
model parameters
$= f$
$t+1$
t
b
Adaptive CA
$\boldsymbol{\sigma}_{t+1} = f(\boldsymbol{\sigma}_t, \boldsymbol{\theta}_t)$
$\boldsymbol{\theta}_{t+1} = g(\boldsymbol{\sigma}_t, \boldsymbol{\theta}_t)$
$= f$
$= g$
$t+1$
t

Figure 1: ***(a)**: Classical CAs are dynamical systems parameterized by fixed update rules that are applied identically at every cell in the grid. (**b**): Adaptive CAs extend the same principles of classical CAs onto the update rule itself, allowing the rule to also change in space and time. The evolution of the update rule can itself behave like a CA, reacting to its neighbors' activities and patterns. Coupling these systems allows one to create and study much more expressive models that simulate complex phenomena such as driven, dissipating, and self-organizing non-equilibrium dynamics that are characteristic of life.*

phases where many biological systems seem to operate, taking advantage of the computational/dynamical properties that characterize these critical points. However, setting a system to a critical point usually requires an accurate fine-tuning of the control parameters involved. Therefore, in order to claim that a biological system is at the edge of criticality, one has to show the existence of self-organizing mechanisms (SoC) that bring the control parameter to a critical point. This has been a common paradigm since the introduction of SOC models by (Bak et al., 1987).

The Ising model has been, for the better part of a century, the foundational model to study the nature and mathematics behind second-order phase transitions (Ising, 1924). Here we simulate a 2D Ising model CA and introduce mechanisms for the model to self-organize to criticality (Onsager, 1944).

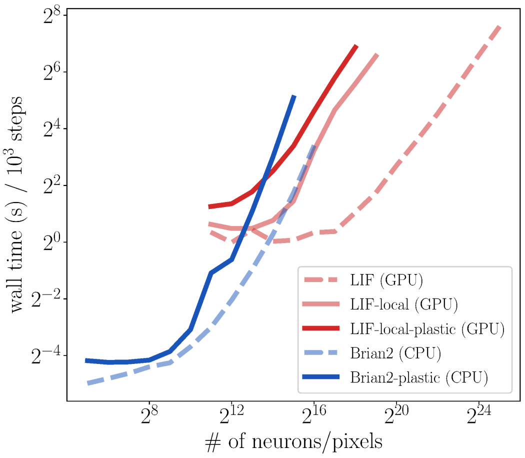


Figure 2: *Benchmark of our leaky integrate-and-fire adaptive CA model with homogeneous kernels (LIF), heterogeneous kernels (LIF-local), and plastic kernels (LIF-local-plastic) running on a single 12GB GTX 2080 compared to neural network simulations run with Brian2 (with and without plasticity) on a 8-core CPU with a single thread. All models were scaled until failure/could not be run on a single computer.*

The Ising model has a set of state variables called spins s_i, which can be in the discrete positions of -1 or $+1$. The spins are connected in a 2D grid using a Von Neumann neighborhood connectivity pattern with common weights J. The system is characterized by its energy landscape, defined as:

$$E(s) = -\sum_{\langle i,j \rangle} J_{ij} s_i s_j. \tag{3}$$

When $J > 0$, spins tend to align to minimize their energy. In order to simulate the system, the Metropolis algorithm is used to update a randomly initialized state to arrive at an equilibrium configuration. An individual spin is flipped towards configurations that minimize its local energy. Flips that decrease the energy ($\Delta E < 0$) are always accepted, while the probability of energetically unfavorable ($\Delta E > 0$) flips are given by

$$p(-s_i) = e^{-\Delta E_i \beta}, \tag{4}$$

where $\beta = 1/(kT)$, k is the Boltzmann constant which we set to 1, and T is the temperature of the model controlling the randomness of the system. At each timestep, a small fraction of the spins are allowed to update according to their flip probabilities. To make the system to self-organize to criticality, we allow the temperature to evolve depending on the system's state so that it can tune itself. Two methods,

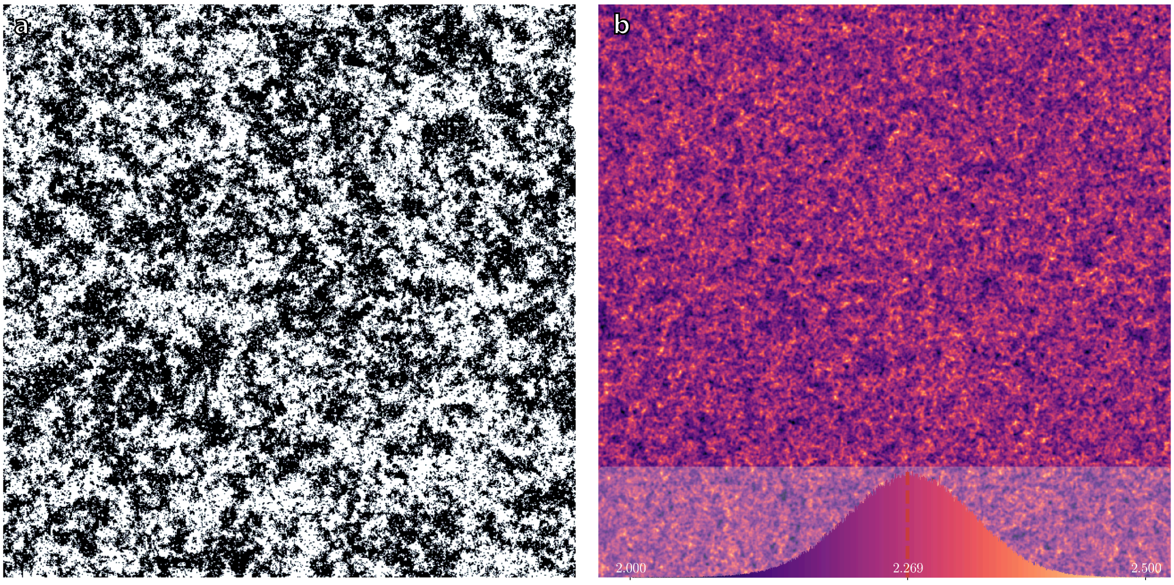


Figure 3: *Spins (**a**) and temperatures (**b**) of a local adaptive Ising model of size 1280x1280 near* $T_c \approx 2.269$. *Overlay in (b) shows the distribution of local temperatures (centered at critical temperature) and indicates the color code.*

a local and a global method, are defined and applied to the model as it updates.

The local adaptation method allows each cell/spin site to take a measurement of its nearest neighbors' magnetization

$$m_i = \frac{1}{n} \sum_{j \in \langle nn \rangle}^{n} s_j \tag{5}$$

and change its temperature according to an update rule that increases the local temperature if a system is too ordered (magnetized). The temperature is also allowed to diffuse, so neighboring cells will slowly average out their differences. The temperature decays at a rate proportional to itself. Combining all terms, growth, decay, and diffusion, we obtain:

$$\Delta T_i = \alpha m_i^2 - \epsilon T_i + D \nabla^2 T_i, \tag{6}$$

where $T_i\,(t+1) = \eta T_i\,(t) + \Delta T_i$. The values $\eta = 0.5$, $\alpha = 1 \times 10^{-1}$, $\epsilon = 2 \times 10^{-2}$, $D = 1$. were used for the local update rule. Pseudo-code of the patch operations required by this local model are summarized in Procedure 3.

Alternatively, the global adaptation method updates the system's global temperature using global magnetization measurements:

$$\Delta T = \alpha |M| - \xi(t) T^2, \tag{7}$$

where the global measurement of the system's magnetization is defined as:

$$|M| = \left| \frac{1}{N} \sum_{j}^{N} s_j \right|. \tag{8}$$

The update follows a similar rule as the local temperature logic: the ordered systems have their temperature increased, and the disordered systems – decreased. A folded normal distribution scaled by T^2 is used to sample noise $\xi(t)$ for the decay term. The parameters of the global update rule share the same coefficients as the local method, with the exception of ϵ, which is further divided by the linear size of the system. With a proper choice of parameters, both methods are capable of self-organizing the Ising model to its known critical temperature, see Fig. 4.

Some difficulties in discovering useful adaptation rules can be explained with the stability analysis of these coupled systems. In the examples we used, a homogeneous, linear stability analysis of a Hohenberg-Halperin model A (a macroscopic Langevin equation for the Ising model) combined with the adaptation equations shows that the critical fixed point might not always be stable, meaning that the mechanism is not truly self-organized. However, establishing a link between the parameters of the simulated model with Glauber dynamics and the macroscopic model is a difficult task, meaning that it is not possible to provide exact boundaries for sets of parameters that allow the model to self-organize. Despite this, we could find adequate param-

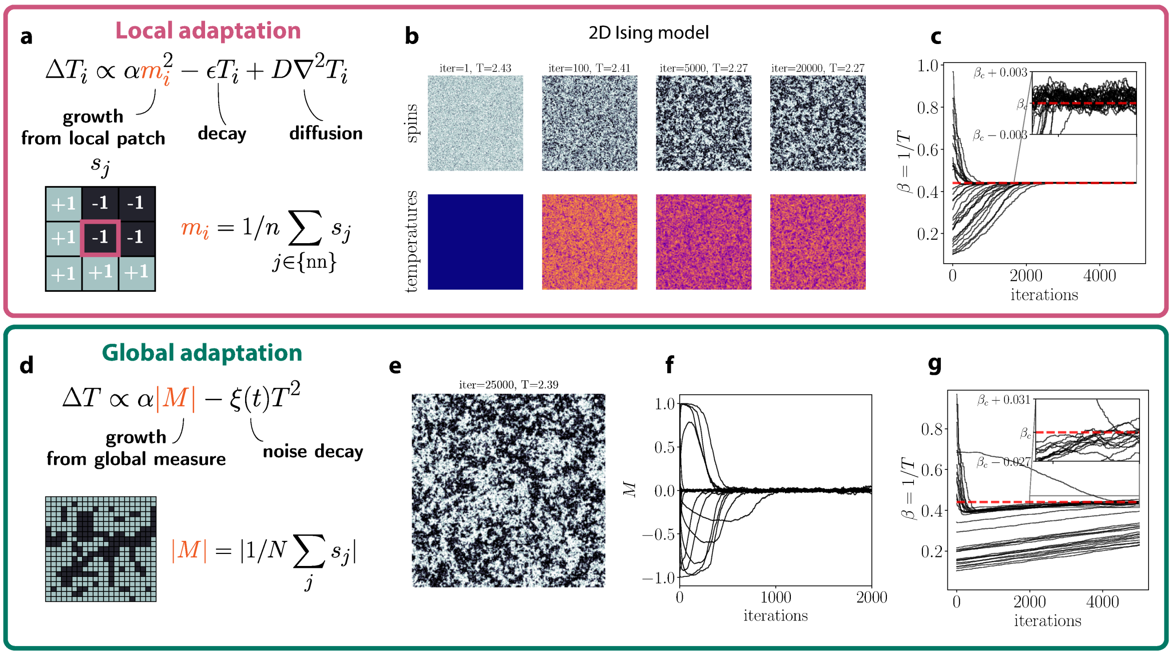


Figure 4: *Two methods to achieve self-organization to criticality in the 2D Ising model: a local/microscopic method (top panels, local temperature evolution described in **a**), and a global/macroscopic method (bottom panels, global temperature update in **d**).**b**: snapshots in time of the spin and temperature.**c**: the convergence of the average local temperature to the known critical temperature of the 2D Ising model ($T_c \approx 2.269$) for a variety of initial temperatures. The global adaptation method was harder to balance and required more time to converge. **e**: A snapshot of spins after 25000 timesteps. The magnetization (**f**) and inverse temperature (**g**) are plotted over time.*

eter values simply by a trial-and-error approach, meaning that it is not complicated to make the model self-organized in practice.

Here, we demonstrate our SoC model on the 2D Ising network because the dynamics of this model are well-studied and drawn out before us, thus, we can verify if our methods perform as we expect them. However, one hopes to understand the broader landscapes of more complex and realistic systems, and that requires experimentation. Navigating the phase diagrams of arbitrary models in a goal-oriented way is likely to be challenging, so we explore these self-organizing models in order to understand via creation and experimentation how one might traverse these high-dimensional spaces with a principled approach.

Procedure 3 SOC 2D-Ising Model

```
function NEWTEMP($spins, temps, \alpha, \epsilon, D$)
    $tPatches \leftarrow$ unfold($temps, patchSize$)
    $sPatches \leftarrow$ unfold($spins, patchSize$)
    for $T_{ij}, S_{ij}$ in zip($tPatches, sPatches$) do
        $m_{ij} \leftarrow$ abs(mean($S_{ij}$))
        $newT_{ij} \leftarrow \alpha m_{ij}^2 - \epsilon T_{ij} + D\nabla^2 T_{ij}$
        $out_{ij} \leftarrow$ smooth($T_{ij}, newT_{ij}$)
    end for
    return $newT$
end function
```

Synaptic plasticity in a neural CA

Simulating biological neural networks of a plausible scale in a reasonable amount of time is a well-established challenge in the field of computational neuroscience (Giannakakis et al., 2020). Given their ability to simulate millions of locally connected units in real time, CAs are a natural candidate for such large-scale simulations. Indeed, it has been recently shown that CAs can be fitted to model the activity of spiking neural networks (Farner et al., 2021). Nevertheless, one of the main features of biological neural networks is the ability of each individual neuron to learn and adapt to external stimuli by modifying its internal parameters and its connectivity to neighboring neurons. This feature of biological neural networks is not possible to capture with traditional CAs, where the update rule of each pixel/neuron is

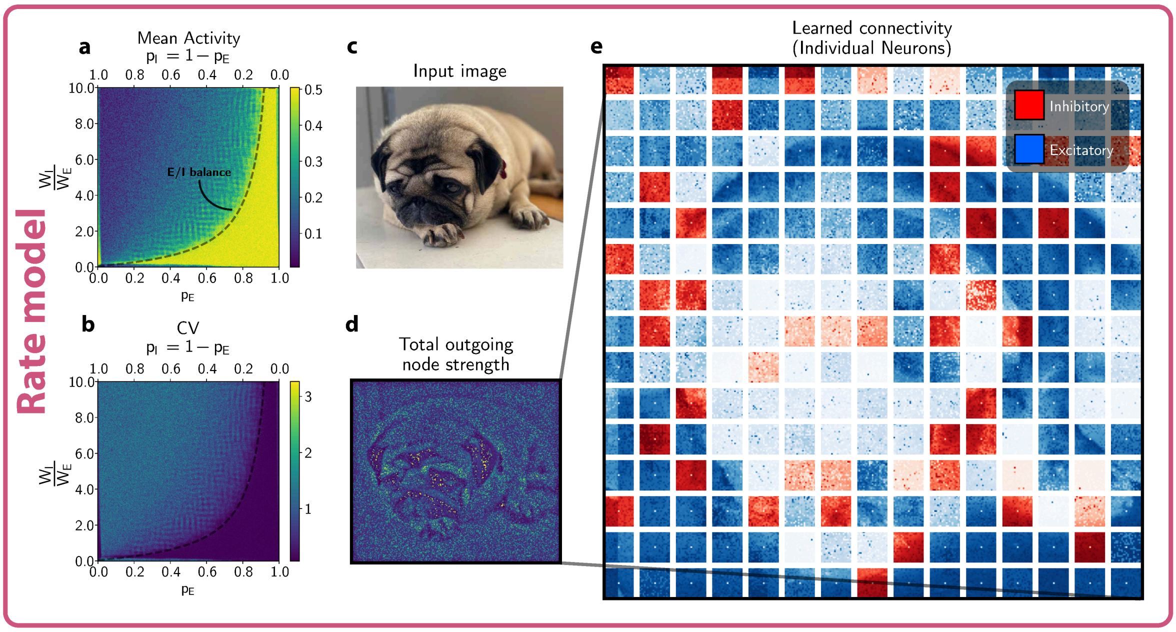


Figure 5: ***(a)**: A phase diagram of the mean activity of the rate model as the excitatory neuronal density (probability of each neuron to be excitatory, p_E is varied from $[0,1]$ on the x-axis, and the relative weights of inhibitory/excitatory connections are varied from $[0,10]$ on the y-axis. The dashed line marks the balance of average incoming excitatory and inhibitory connections to each neuron. **(b)**: The coefficient of variation (std./mean) for the same range of parameters as **(a)**. **(c)**: A natural image is given as input to the network, and the connectivity after the convergence of plasticity **(d)**. **(e)**: The incoming weights matrices of 15x15 example neurons are shown, with inhibitory/excitatory neurons colored red/blue.*

the same as with all others and remains constant in time. The adaptive update rules can solve this problem by enabling the modeling of local threshold adaptation as well as synaptic plasticity in neural CAs.

Rate model: We first create a rate network consisting of neurons following Wilson-Cowan dynamics (Wilson and Cowan, 1972) arranged in a 2D grid. Each neuron's activity is given by:

$$\tau\frac{dr}{dt} = -r + (1-r)\cdot s(WR + c), \tag{9}$$

where, s is a sigmoid function, W the neuron's connectivity matrix, R the activity of neighboring neurons and c a constant current.

Each neuron can be either excitatory or inhibitory and is connected to its neighboring neurons within a given radius that varies between simulations.

We model distinct plasticity mechanisms active on the excitatory and inhibitory synapses (i.e., connections of the neuron follow a different plasticity rule depending on its type). Specifically, the weight of the connection from the excitatory neuron i to neuron j is modified according to a simple Hebbian rule (Hebb, 1949):

$$\Delta W_{ij}^{E} = r_i \cdot r_j. \tag{10}$$

While the connection weight from the inhibitory neuron i to neuron j is modified according to a homeostatic inhibitory rule (Vogels et al., 2011):

$$\Delta W_{ij}^{I} = r_i \cdot (r_j - \beta), \tag{11}$$

where β is a target activity level. To further stabilize learning, we use synapse-type specific normalization that has been demonstrated to produce stable representations of inputs as well as E/I co-tuning in rate and spiking neural networks (Eckmann and Gjorgjieva, 2023; Giannakakis et al., 2023). The resulting network displays rich E/I dynamics, Fig. 5a, b.

Including this combination of plasticity mechanisms in neural CAs, allows us to study phenomena such as assembly formation (Miehl et al., 2022) on a much larger scale than has ever been attempted before. In this study, we investigate whether natural images can be imprinted in the

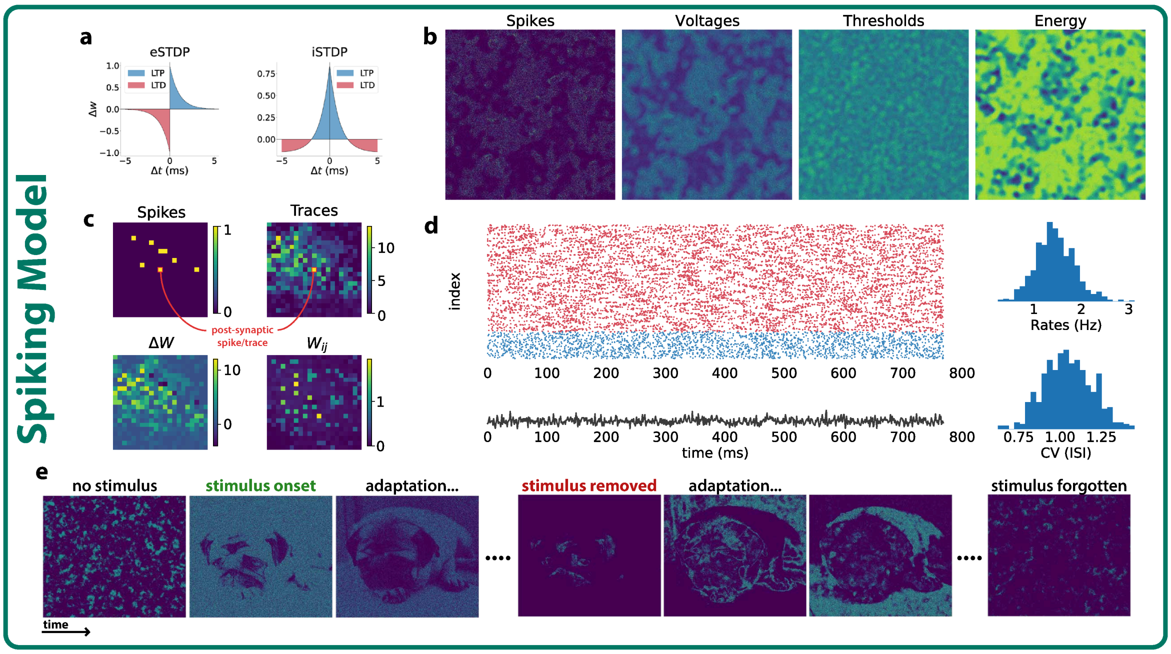


Figure 6: ***a.*** *The synaptic plasticity rules for excitatory (left) and inhibitory (right) synapses.* ***b.*** *The activity of different channels of the CA. Each channel tracks different biophysical aspects of the neuron and is updated following different update rules.* ***c.*** *The pre-synaptic spikes and plasticity traces of a random neuron at location i, j are shown, where the central pixel represents the post-synaptic neuron's spikes/traces. Applying the plasticity rule to this neuron, the connectivity is modified by ΔW, and the resulting normalized connectivity W_{ij} is shown in the bottom row.* ***d.*** *The network's spiking activity is asynchronous irregular (a widely observed state in biological neural networks), with a broad distribution of firing rates. The distribution of the CVs of the inter-spike intervals centered around 1, indicating a Poisson-like activity.* ***e.*** *Threshold adaptation enables short-term memory in our model. Once the stimulus is presented, the activity quickly adapts to the statistics of the presented image. The inverted outline of the image persists in the network's activity for a short period after the stimulus is removed due to the relatively slow timescales of threshold adaptation.*

network's connectivity via synaptic plasticity. We project a natural image as additional incoming current c each neuron receives (each neuron receives the grayscale value of a single pixel as input). Once the plasticity converges, we examine the learned connectivity. We plot (Fig. 5d, e) the learned connectivity matrix of the network by coloring every neuron/pixel proportional to its out/in-going weights, respectively, showing that the input stimuli has been imprinted in the model's connectivity. Our results demonstrate that natural images can be consistently imprinted in our network's connectivity via a combination of simple plasticity mechanisms.

Spiking model: We then create a detailed simulation of a spiking neural network. We model different aspects of the neuron's behavior with different CA channels, each representing the local properties of a single cell and interacting with each other according to known differential equations.

The primary channel of the model tracks the membrane voltage of each neuron:

$$\tau \frac{dV}{dt} = -V + I_e + c, \tag{12}$$

where I_e is the input coming from other neurons and c is a constant baseline current. To demonstrate that we can use CAs to model neurons at an arbitrary level of biological accuracy, we use eLIF neurons (leaky integrate and fire neurons with an additional energy constraint) (Fardet and Levina, 2020), a model that demonstrates a high degree of biological plausibility, Fig. 6 b. Thus, a neuron spikes when the voltage goes above a threshold V_{th} and its energy is sufficiently high:

$$S_t = \begin{cases} 1, & \text{if } V \geq V_{th} \text{ and } E \geq E_{min}, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

where S is the CA channel tracking spikes, E is the neuron's energy and E_{min} is the minimum energy required for spiking.

The energy E is tracked by a different channel and follows the equation:

$$\tau_e \frac{dE}{dt} = \epsilon \cdot (E_0 - E) - s_c \cdot \sum_f \delta(t - t^f), \tag{14}$$

where ϵ is an energy replenishment rate, E_0 the target energy, s_c the energetic cost of a spike and t_f the neuron's spike times ($t_f = \{t : S_t = 1\}$).

Moreover, each neuron's spiking threshold is modified by threshold adaptation, a homeostatic plasticity mechanism associated with increased robustness in spiking networks (Fontaine et al., 2014; Huang et al., 2016). Thus, a separate CA channel tracks the spike threshold, which changes according to the following equation:

$$\tau_{th} \frac{dV_{th}}{dt} = -V_{th} + \eta_{th} \cdot (A - \rho_0), \tag{15}$$

where η_{th} is an adaptation rate, ρ_0 is a target firing rate, and A is the neuron's spike trace, which is represented by a final CA channel as:

$$\tau_a \frac{dA}{dt} = -A + \sum_f \delta(t - t^f). \tag{16}$$

The multiple CA channels allow an arbitrary degree of biological plausibility, with more details being potentially included by adding new channels. For example, more detailed voltage equations, refractoriness, and other details can be added if necessary.

We find that the spiking network exhibits realistic population dynamics. In particular, we see that it self-organizes in an asynchronous irregular firing state, a commonly observed dynamical regime in spiking networks (Brunel, 2000), see Fig. 6 d.

Additionally, we examine whether it can encode natural images similar to the simplified rate network. Indeed, we find that synaptic plasticity allows the encoding of images in the network's connectivity. Finally, the inclusion of threshold adaptation enables a short-term preservation of image statistics in the network's spiking activity once the stimulus has been removed. This observation is consistent with other findings about the relationship between short-term memory and threshold adaptation (Itskov et al., 2011; Hu et al., 2021) and suggests that our model can replicate different key properties of spiking networks.

Scalability and computational efficiency

We tested the performance of our method compared to the standard neuronal activity simulator Brian2 (Stimberg et al., 2019). On a single machine, our method allows simulating networks that are two orders of magnitude larger, where simulation time is scaling particularly favorably for large networks, Fig. 2. Furthermore, the methods demonstrated in this paper are generalizable to a much broader variety of dynamical systems than most specialized and highly-optimized simulators of collective systems without too much loss of efficiency, can run and be visualized live, and can be made interactive with human intervention. We believe all these properties are essential for building deeper understandings of these emergent systems.

Conclusion

For the scientific study of the emergence and collective behavior in natural systems to become plausible in a systematic way, we first need to be able to reliably recreate and analyze such phenomena in simpler and well-understood artificial systems. Cellular automata are a natural candidate system for studying this kind of emergent behavior in a controlled and simplified manner. Here, we explore methods to expand the capabilities of CAs to model complex phenomena by introducing heterogeneity in their update rules.

Enabling CAs to be heterogeneous massively increases their expressivity. In fact, accurately describing any truly complex natural system requires some kind of heterogeneity in the modeling. Here, we demonstrate how the inclusion of such heterogeneity in the update rules of CAs can enable the modeling of complex natural systems in real-time on a very large scale. We use several examples of large-scale complex systems (Ising model, biological neural networks) that can utilize heterogeneous update rules to self-organize towards a desired state (critical point in the case of the Ising model, natural image derived attractor state in the case of the rate and spiking neural networks). As a follow-up example, one can study the self-organization of critical-like dynamics in neuronal networks with synaptic plasticities (Zeraati et al., 2021), or using the scalability of the system to previously impossible sizes to study the scaling behavior of the local networks (Zeraati et al., 2022).

Our approach is not limited to the specific systems we chose to model but is a general method that can be extended to many other complex, self-organized systems that require heterogeneous update rules. Any system with some kind of bottom-up organizational principle (i.e., locally interacting units) can potentially be modeled using locally adaptive CAs. Thus, we propose that adaptive CAs can be a useful tool to discover, either through experimentation or optimization, adaptation mechanisms that can self-organize dynamical systems in goal-oriented ways.

Acknowledgements

This work was supported by a Sofja Kovalevskaja Award from the Alexander von Humboldt Foundation. EG and SK thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for their support. We acknowledge the support from the BMBF through the Tübingen AI Center (FKZ: 01IS18039A). AL is a member of the Machine Learning Cluster of Excellence, EXC number 2064/1 – Project number 39072764.

References

Bak, P., Tang, C., and Wiesenfeld, K. (1987). Self-organized criticality: An explanation of the 1/f noise. *Physical review letters*, 59(4):381.

Brunel, N. (2000). Dynamics of sparsely connected networks of excitatory and inhibitory spiking neurons. *Journal of Computational Neuroscience*, 8.

Chan, B. W.-C. (2020). Lenia and expanded universe. In *ALIFE 2020: The 2020 Conference on Artificial Life*, pages 221–229. MIT Press.

Chopard, B. and Droz, M. (1998). Cellular automata. *Modelling of Physical*.

Eckmann, S. and Gjorgjieva, J. (2023). Synapse-type-specific competitive hebbian learning forms functional recurrent networks. *bioRxiv*.

Fardet, T. and Levina, A. (2020). Simple models including energy and spike constraints reproduce complex activity patterns and metabolic disruptions. *PLOS Computational Biology*, 16(12):1–22.

Farner, J. J., Weydahl, H., Jahren, R., Ramstad, O. H., Nichele, S., and Heiney, K. (2021). Evolving spiking neuron cellular automata and networks to emulate in vitro neuronal activity. *CoRR*, abs/2110.08242.

Fontaine, B., Peña, J. L., and Brette, R. (2014). Spike-threshold adaptation predicted by membrane potential dynamics in vivo. *PLOS Computational Biology*, 10(4):1–11.

Giannakakis, E., Han, C. E., Weber, B., Hutchings, F., and Kaiser, M. (2020). Towards simulations of long-term behavior of neural networks: Modeling synaptic plasticity of connections within and between human brain regions. *Neurocomputing*, 416:38–44.

Giannakakis, E., Vinogradov, O., Buendia, V., and Levina, A. (2023). Recurrent connectivity structure controls the emergence of co-tuned excitation and inhibition. *bioRxiv*.

Gilpin, W. (2019). Cellular automata as convolutional neural networks. *Physical Review E*, 100(3):032402.

Hebb, D. O. (1949). The first stage of perception: growth of the assembly. *The Organization of Behavior*, 4(60):78–60.

Hernandez, A., Vilalta, A., and Moreno-Noguer, F. (2021). Neural cellular automata manifold. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10020–10028.

Hu, B., Garrett, M. E., Groblewski, P. A., Ollerenshaw, D. R., Shang, J., Roll, K., Manavi, S., Koch, C., Olsen, S. R., and Mihalas, S. (2021). Adaptation supports short-term memory in a visual change detection task. *PLOS Computational Biology*, 17(9):1–22.

Huang, C., Resnik, A., Celikel, T., and Englitz, B. (2016). Adaptive spike threshold enables robust and temporally precise neuronal encoding. *PLOS Computational Biology*, 12(6):1–25.

Ising, E. (1924). *Beitrag zur theorie des ferro-und paramagnetismus*. PhD thesis, Grefe & Tiedemann Hamburg.

Itskov, V., Curto, C., Pastalkova, E., and Buzsáki, G. (2011). Cell assembly sequences arising from spike threshold adaptation keep track of time in the hippocampus. *Journal of Neuroscience*, 31(8):2828–2834.

Langton, C. G. (1986). Studying artificial life with cellular automata. *Physica D: Nonlinear Phenomena*, 22(1-3):120–149.

Louis, P.-Y. and Nardi, F. R. (2018). Probabilistic cellular automata. *Emergence, Complexity, Computation*, 27.

McCabe, J. (2010). Cyclic symmetric multi-scale turing patterns. In *Proceedings of Bridges 2010: Mathematics, Music, Art, Architecture, Culture*, pages 387–390.

Miehl, C., Onasch, S., Festa, D., and Gjorgjieva, J. (2022). Formation and computational implications of assemblies in neural circuits. *The Journal of Physiology*.

Mordvintsev, A., Randazzo, E., Niklasson, E., and Levin, M. (2020). Growing neural cellular automata. *Distill*. https://distill.pub/2020/growing-ca.

Onsager, L. (1944). Crystal statistics. i. a two-dimensional model with an order-disorder transition. *Physical Review*, 65(3-4):117.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Pezzulo, G. and Levin, M. (2016). Top-down models in biology: explanation and control of complex living systems above the molecular level. *Journal of The Royal Society Interface*, 13(124):20160555.

Pontes-Filho, S., Walker, K., Najarro, E., Nichele, S., and Risi, S. (2022). A single neural cellular automaton for body-brain co-evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '22, page 148–151, New York, NY, USA. Association for Computing Machinery.

Rampe, J. (2021). Multi-scale turing patterns. https://softologyblog.wordpress.com/2011/07/05/multi-scale-turing-patterns/.

Randazzo, E., Mordvintsev, A., Niklasson, E., Levin, M., and Greydanus, S. (2020). Self-classifying mnist digits. *Distill*. https://distill.pub/2020/selforg/mnist.

Slackermanz (2021). Understanding multiple neighborhood cellular automata. https://slackermanz.com/understanding-multiple-neighborhood-cellular-automata/.

Stimberg, M., Brette, R., and Goodman, D. F. (2019). Brian 2, an intuitive and efficient neural simulator. *Elife*, 8:e47314.

Toffoli, T. (1984). Cellular automata as an alternative to (rather than an approximation of) differential equations in modeling physics. *Physica D: Nonlinear Phenomena*, 10(1-2):117–127.

Turing, A. (1952). The chemical basis for morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, 237(641):37–72.

Variengien, A., Nichele, S., Glover, T., and Pontes-Filho, S. (2021). Towards self-organized control: Using neural cellular automata to robustly control a cart-pole agent. *arXiv preprint arXiv:2106.15240*.

Vichniac, G. Y. (1984). Simulating physics with cellular automata. *Physica D: Nonlinear Phenomena*, 10(1-2):96–116.

Vogels, T. P., Sprekeler, H., Zenke, F., Clopath, C., and Gerstner, W. (2011). Inhibitory plasticity balances excitation and inhibition in sensory pathways and memory networks. *Science*, 334(6062):1569–1573.

von Neumann, J. (1966). The theory of self-reproducing automata, burks aw (ed). urbana and london.

Wilson, H. and Cowan, J. (1972). Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysical journal*, 12(1):1—24.

Wolfram, S. (1984). Cellular automata as models of complexity. *Nature*, 311(5985):419–424.

Zeraati, R., Buendía, V., Engel, T. A., and Levina, A. (2022). Topology-dependent coalescence controls scaling exponents in finite networks. *arXiv preprint arXiv:2211.06296*.

Zeraati, R., Priesemann, V., and Levina, A. (2021). Self-organization toward criticality by synaptic plasticity. *Frontiers in Physics*, 9:103.

A.4 Emergent mechanisms for long timescales depend on training curriculum and affect performance in memory tasks

Emergent mechanisms for long timescales depend on training curriculum and affect performance in memory tasks

Sina Khajehabdollahi ^{1,2,*}**, Roxana Zeraati** ^{1,3,*}**, Emmanouil Giannakakis** ^{1,3}**,**
Tim J. Schäfer ^{1,3}**, Georg Martius** ^{1,2}**, Anna Levina** ^{1,3}
^{1} University of Tübingen, Germany
^{2} Max Planck Institute for Intelligent Systems, Tübingen, Germany
^{3} Max Planck Institute for Biological Cybernetics, Tübingen, Germany
* These authors contributed equally to this work.
{firstname.lastname}@uni-tuebingen.de

Abstract

Recurrent neural networks (RNNs) in the brain and *in silico* excel at solving tasks with intricate temporal dependencies. Long timescales required for solving such tasks can arise from properties of individual neurons (single-neuron timescale, τ, e.g., membrane time constant in biological neurons) or recurrent interactions among them (network-mediated timescale, τ_{net}). However, the contribution of each mechanism for optimally solving memory-dependent tasks remains poorly understood. Here, we train RNNs to solve N-parity and N-delayed match-to-sample tasks with increasing memory requirements controlled by N, by simultaneously optimizing recurrent weights and τs. We find that RNNs develop longer timescales with increasing N, but depending on the learning objective, they use different mechanisms. Two distinct curricula define learning objectives: sequential learning of a single-N (single-head) or simultaneous learning of multiple Ns (multi-head). Single-head networks increase their τ with N and can solve large-N tasks, but suffer from catastrophic forgetting. However, multi-head networks, which are explicitly required to hold multiple concurrent memories, keep τ constant and develop longer timescales through recurrent connectivity. We show that the multi-head curriculum increases training speed and stability to perturbations, and allows generalization to tasks beyond the training set. This curriculum also significantly improves training GRUs and LSTMs for large-N tasks. Our results suggest that adapting timescales to task requirements via recurrent interactions allows learning more complex objectives and improves the RNN's performance.

1 Introduction

The interaction of living organisms with their environment requires the concurrent processing of signals over a wide range of timescales, from short timescales of coding sensory stimuli (Bathellier et al., 2008; Panzeri et al., 2010; Safavi et al., 2023) to longer timescales of cognitive processes like working memory (Jonides et al., 2008). The diverse timescales of these tasks are reflected in the dynamics of the neural populations performing the corresponding computations in the brain (Murray et al., 2014; Cavanagh et al., 2020; Gao et al., 2020; Zeraati et al., 2022). At the same time, artificial neural networks performing memory-demanding tasks (speech (Graves et al., 2013), handwriting (Graves, 2013), sketch (Ha & Eck, 2018), language (Bowman et al., 2015), time series prediction (Chung et al., 2014; Torres et al., 2021), music composition (Boulanger-Lewandowski et al., 2012)) need to process the temporal dependency of sequential data over variable timescales. Recurrent neural networks (RNNs) (Elman, 1990; Hochreiter & Schmidhuber, 1997; Lipton et al., 2015; Yu et al., 2019) have been introduced as a tool that can learn such temporal dependencies using back-propagation through time.

In biological networks, diverse neural timescales emerge via a variety of interacting mechanisms. Timescales of individual neurons in the absence of recurrent interactions are determined by cellu-

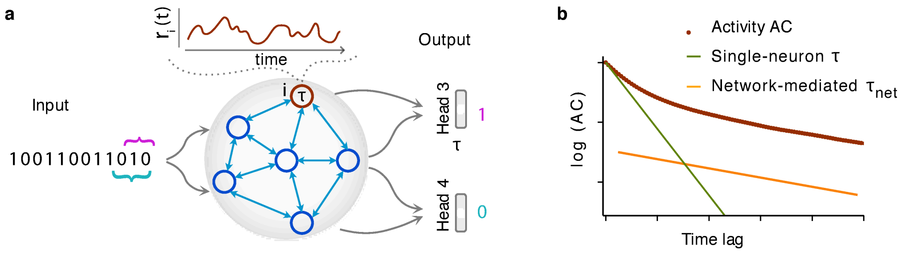


Figure 1: Schematics of network structure and timescales. **a.** An outline of the network. A binary sequence is given as input to a leaky RNN, with each neuron's intrinsic timescale being a trainable parameter τ. The illustration shows the N-parity task with readout heads for $N = 3$ and $N = 4$. **b.** An illustration of the manifestation of different timescales (single-neuron and network-mediated) on the autocorrelation (AC) of a network neuron (see also Fig. S2).

lar and synaptic processes (e.g., membrane time constant) that vary across brain areas and neuron types (Gjorgjieva et al., 2016; Duarte et al., 2017). However, recurrent interactions also shape neural dynamics introducing network-mediated timescales. The strength (Ostojic, 2014; Chaudhuri et al., 2015; van Meegen & van Albada, 2021) and topology (Litwin-Kumar & Doiron, 2012; Chaudhuri et al., 2014; Zeraati et al., 2023; Shi et al., 2023) of recurrent connections give rise to network-mediated timescales that can be much longer than single-neuron timescales.

Heterogeneous and tunable single-neuron timescales have been proposed as a mechanism to adapt the timescale of RNN dynamics to task requirements and improve their performance (Perez-Nieves et al., 2021; Tallec & Ollivier, 2018; Quax et al., 2020; Yin et al., 2020; Fang et al., 2021; Smith et al., 2023b; Jain et al., 2020). In these studies, the time constants of individual neurons are trained together with network connectivity. For tasks with long temporal dependencies, the distribution of trained timescales becomes heterogeneous according to the task's memory requirements (Perez-Nieves et al., 2021). Explicit training of single-neuron timescales improves network performance in benchmark RNN tasks in rate (Tallec & Ollivier, 2018; Quax et al., 2020) and spiking (Yin et al., 2020; Fang et al., 2021; Perez-Nieves et al., 2021) networks and leads to greater robustness (Perez-Nieves et al., 2021) and adaptability to novel stimuli (Smith et al., 2023b). While these studies propose the adaptability of single-neuron timescales as a mechanism for solving time-dependent tasks, the exact contribution of single-neuron and network-mediated timescales in solving tasks is unknown.

Here, we study how single-neuron and network-mediated timescales shape the dynamics and performance of RNNs trained on long-memory tasks. We show that the contribution of each mechanism in solving such tasks largely depends on the learning objective defined by the curriculum. Challenging common beliefs in the field, we identify settings where trainable single-neuron timescales offer no advantage in solving temporal tasks. Instead, adapting RNNs' timescales using network-mediated mechanisms improves training speed, stability and generalizability.

2 Model

We approximate the effect of the membrane timescale of biological neurons by equipping each RNN-neuron with a trainable leak parameter τ, defining the single-neuron timescale (Fig. 1a). The activity of each neuron evolves over discrete time steps t governed by:

$$r_i(t) = \left[\left(1-\frac{\Delta t}{\tau_i}\right)\cdot r_i(t-\Delta t) + \frac{\Delta t}{\tau_i}\cdot\left(\sum_{j\neq i} W_{ij}^R \cdot r_j(t-\Delta t) + W_i^I \cdot S(t) + b^R + b^I\right)\right]_\alpha , \tag{1}$$

where $[\cdot]_\alpha$ is the leaky ReLU function with negative slope α, given by:

$$[x]_\alpha = \begin{cases} x, & x \geq 0 \\ \alpha \cdot x, & x < 0. \end{cases} \tag{2}$$

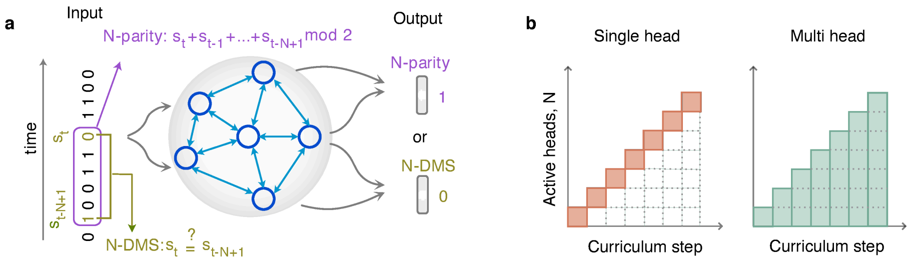


Figure 2: Schematic description of the tasks and curricula **a.** An outline of the network and tasks. In both tasks, the network receives a binary input sequence, one bit at each time step. **b.** In the single-head curriculum, only one read-out head is trained at each curriculum step, while in the multi-head curriculum, a new read-out head is added at each step without removing the older heads.

For all networks, we use $\alpha = 0.1$. We obtain similar results using a different type or location of nonlinearity (Appendix A). W^R, b^R, and W^I, b^I are the recurrent and input weights and biases, respectively, S is the binary input given to the network at each time step, and $\tau_i \geq 1$ is the trainable timescale of the neuron. Unless otherwise stated, the time step is $\Delta t = 1$ (other Δt discussed in Appendix B). Each RNN has 500 neurons. When $\tau = 1$, a neuron becomes memory-less (in isolation) as its current state does not directly depend on its past activity, i.e., memory can only be stored at the network level via interactions. In contrast, for $\tau > 1$, the neuron's activity depends on its past activity, and the dependency increases with τ. In the limit of $\tau \to \infty$, the neuron's activity is constant, and the input has no effect.

The dynamics of each neuron can be characterized by two distinct timescales: (i) single neuron timescale τ, (ii) network mediated timescale τ_{net}. τ gives the intrinsic timescale of a neuron in the absence of any network interaction, while τ_{net} is shaped by the combination of τ and the learned connectivity and represents the effective timescale of the neuron's activity within the network. τ_{net} is generally a function of τ and recurrent weights: $\tau_{\text{net}} = f(\tau, W^R)$(Ostojic, 2014; Chaudhuri et al., 2014; Shi et al., 2023) and $\tau_{\text{net}} \geq \tau$ (Shi et al., 2023). For networks with linear dynamics, τ_{net} can be directly estimated from the eigenvalues of the connectivity matrix normalized by τ (Chaudhuri et al., 2014). For nearest-neighbor connectivity or mean-field dynamics, it is also possible to derive τ_{net} analytically for nonlinear networks. However, a general analytical solution does not exist. Instead, τ_{net} can be effectively estimated from the decay rate of the autocorrelation (AC) function. The AC is defined as the correlation coefficient between the time series and its copy, shifted by time t', called the time lag. For the neuron's activity, it can be computed as

$$\mathrm{AC}_i(t') = \frac{1}{\hat{\sigma_i}^2 (T - t')} \sum_{t=0}^{T-t'} \left(r_i(t) - \hat{\mu}_i\right)\left(r_i(t - t') - \hat{\mu}_i\right), \tag{3}$$

where $\hat{\mu}_i$ and $\hat{\sigma}_i^2$ are the sample mean and variance of $r_i(t)$. To estimate τ_{net}, we drive the network by uncorrelated binary inputs sampled from a Bernoulli distribution.

The AC of a neuron's activity, defined by Eq. 1, can be approximated by two distinct timescales which appear as two slopes in logarithmic-linear coordinates (Fig. 1b) (Shi et al., 2023). The steep initial slope indicates τ, and the shallower slope indicates τ_{net}. In the same way, we characterize the timescale of collective network dynamics by computing population activity (summed activity of all neurons within a network) timescale τ_{pop}, which reflects the timescale of network dynamics as a whole. To avoid AC bias in our estimates (Zeraati et al., 2022), we use long simulations ($T = 10^5$ time steps). We simulate each network for 10 trials (i.e. 10 distinct realizations of inputs) and compute the average AC of each neuron across trials. To estimate τ_{net}, we fit the average AC with a single- ($\tau_{\text{net}} = \tau$) and with a double-exponential ($\tau_{\text{net}} > \tau$) decay function using the nonlinear least-squares method. Then, we use the Akaike Information Criterion (AIC) (Akaike, 1974) to select the best-fitting model. For most neurons (above 95%), Bayesian information criterion (BIC) selects the same model (Fig. S1) and previous work (Pasula, 2023) indicates that AIC provides similar results as the sum of three information criteria (AIC, BIC and Hannan-Quinn information criteria).

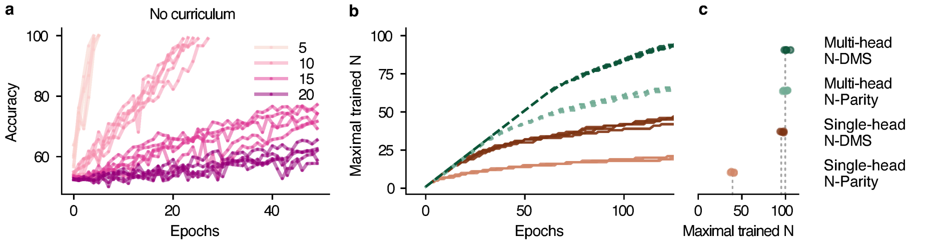


Figure 3: Training performance depends on the curriculum. **a.** Accuracy of training the networks (N-parity task) without a curriculum increases slowly, especially when $N > 10$. For each N, 5 models are independently trained for 50 epochs or until reaching $> 98\%$ accuracy. **b.** Multi-head (dashed) trained networks are solving larger Ns than single-head (solid) within the same training time (colors in c). **c.** The maximum trained N for each task/curriculum at the end of training (1000 epochs or solving $N = 101$, whichever comes first). Gray lines - mean value across 4 networks.

When the double-exponential model is selected, the slowest of two timescales indicates τ_{net}. For most fits, we obtain a large coefficient of determination, confirming a good quality of fit (Fig. S2).

3 Setup

3.1 Tasks

In both tasks (Fig. 2a), a binary sequence S is given as the input, one bit at each time step. We train the networks on sequences with lengths uniformly chosen from the interval $L \in \{N + 2, 4N\}$.
N-delayed match-to-sample (N-DMS): The network outputs 1 or 0 to indicate whether the digit presented at current time t matches the digit presented at time $t - N + 1$. To update the output at every time step, the network needs to store the values and order of the last N digits in memory.
N-parity: The network outputs the binary sum (XOR) of the last N digits. N-parity has a similar working memory component as N-DMS, but requires additional computations (binary sum).

3.2 Training

We train single-neuron timescales $\tau = \{\tau_1, \ldots, \tau_{500}\}$, W^R, b^R, W^I, b^I, and a linear readout layer via back-propagation through time using a stochastic gradient descent optimizer with Nesterov momentum and a cross-entropy loss. Each RNN is trained on a single Nvidia GeForce 2080ti for 1000 epochs, 3 days, or until the $N = 101$ task is solved, whichever comes first. RNNs are trained without any regularization. Including L2 regularization achieves comparable performance.

Single-head: Starting with $N = 2$, we train the network to reach an accuracy of 98%. We then use the trained network parameters to initialize the next network that we train for $N + 1$ (Fig. 2b).
Multi-head: As with the single-head networks, we begin with a network solving a task for $N = 2$, but once a threshold accuracy of 98% is reached, a new readout head is added for solving the same task for $N + 1$, preserving the original readout heads. At each curriculum step, all readout heads are trained simultaneously (the loss is the sum of all readout heads' losses) so that the network does not forget how to solve the task for smaller Ns (Fig. 2b).

4 Results

4.1 Performance under different curricula

Necessity of curriculum: Our objective is to learn the largest possible N in each task. First, we test whether a good performance can be achieved for high N in either task without any curricula. We find that for both tasks (see Appendix E for N-DMS results), networks struggle to reach high accuracy for $N > 10$ (Fig. 3a, Fig. S13). However, using either curriculum significantly boosts the network's capacity to learn tasks with larger N (Fig. 3b).

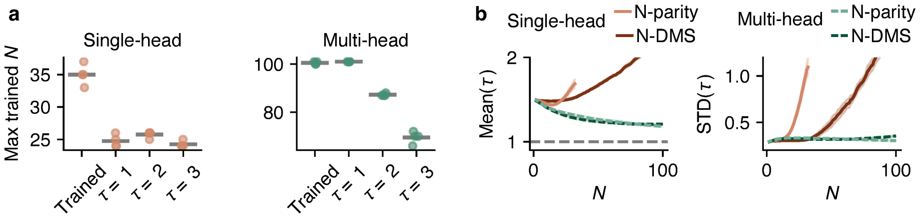


Figure 4: Importance of single-neuron timescales for different curricula. **a.** The maximum N solved in the N-bit parity task after 1000 epochs (reaching an accuracy of 98%). X-label indicates training constraints: $\tau = 1, 2$ or 3 - fixed τ with only weights being trained, "Trained" allows training of τ. In the single-head curriculum, models rely on training τ, whereas in the multi-head curriculum, τ fixed $\tau = 1$ is as good as training τ. Horizontal bars - mean. **b.** The mean and standard deviation (STD) of the trained τs increase with N in single-head networks. In contrast, in multi-head networks, the mean τ decreases towards 1, and the STD remains largely constant. The mean and STD are computed across neurons within each network (up to the maximum N shared between all trained networks). Shading - variability across 4 trained networks.

Comparison of curricula: Task performance differs significantly between curricula. The single-head networks can reliably reach $N \approx 35$ for the N-parity and $N \approx 90$ for the N-DMS task. The difference in performance between the two tasks is expected because the N-DMS task is much easier than the N-parity task. However, the multi-head networks can reliably reach $N \geq 100$ for both tasks and require fewer training epochs to reach 98% accuracy for each N (Fig. 3b, c). Networks that are trained using an intermediate curriculum between the two extremes of single- and multi-head (i.e., solving simultaneously $H < N$ tasks for $N, N-1, \ldots, N-H+1$ with gradually increasing Ns) exhibit an intermediate performance (Appendix F). Overall, the multi-head networks outperform the single-head ones in terms of performance (maximum N reached) and the required training time. Moreover, the multi-head curriculum significantly improves the training of other recurrent architectures such as GRU and LSTM to perform large-N tasks (Appendix G), suggesting that the multi-objective curriculum can generally improve learning long-memory tasks.

The superior performance of the multi-head networks may be counter-intuitive since they solve the task for every $N \leq m$ at the m-th step of the curriculum, whereas the single-head networks only solve it for exactly $N = m$. However, we find that the single-head networks suffer from catastrophic forgetting: a network trained for larger N cannot perform the task for smaller Ns it was trained for, even after retraining the readout weights (Appendix H). These results suggest that explicit prevention of catastrophic forgetting by learning auxiliary tasks (i.e. tasks with N smaller than objective, $N < m$) facilitates learning large Ns. Interestingly, training directly on a multi-N task without using an explicit curriculum results in the emergence of the multi-head curriculum: networks learn to first solve small-N tasks and then large-N tasks (Appendix I), supporting the use of the multi-head step-wise strategy.

Necessity of training τ: We examine the impact of training single-neuron timescales on training performance. We compare the training performance of networks with a fixed $\tau \in \{1, 2, 3\}$ shared across all neurons versus networks with trainable timescales. In the single-head curriculum, the training performance with fixed τ is significantly worse than when we train τ (Fig. 4a). On the other hand, In multi-head networks, training performance is the same for fixed $\tau = 1$ and trainable τ cases, but steeply declines for fixed $\tau \geq 2$ (Fig. 4b). See (Fig. S10) for N-DMS results. These results indicate that single-head networks rely on τ for solving the task, whereas multi-head networks only use τ to track the timescale of updating the input and rely on other mechanisms to hold the memory.

4.2 Mechanisms underlying long timescales

To uncover the mechanisms underlying the difference between the two curricula, we study how τ, τ_{net}, τ_{pop} and recurrent weights change with increasing task difficulty N. One can expect that as the timescale of the task (mediated by N) increases, neurons would develop longer timescales to integrate the relevant information. Such long timescales can arise either by directly modulating τ for each neuron or through recurrent interactions between neurons reflected in τ_{net} and τ_{pop}.

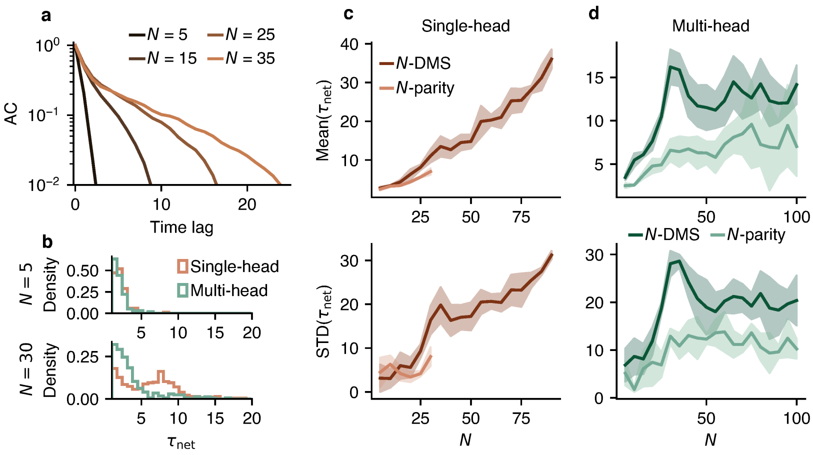


Figure 5: The emergence of network-mediated timescales depends on the curriculum. **a.** Example average ACs of all the neurons within a single-head network, N-parity task. The ACs of individual neurons' activity decay slower with increasing N. **b.** Distributions of the network-mediated timescales τ_{net} for single and multi-head networks solving N-parity task for $N = 5$ and $N = 30$. The distribution becomes broader for higher N. **c, d.** The mean and STD of the network-mediated timescale τ_{net} increase with N in both tasks. The mean and STD are computed across neurons within each network. Shades - variability across 4 trained networks.

Dependence of τ on N: The two curricula adjust their τ to N in distinct ways: single-head networks increase their τ with N, but multi-head networks prefer $\tau \to 1$ (Fig. 4b). For the single-head curriculum, the mean and variance of τ increase with N, suggesting that not only τs become longer as the memory requirement grows, but they also become more heterogeneous. We obtain similar results for networks trained without curriculum (Fig. S3). On the contrary, in multi-head networks, the average τ decreases with N, approaching $\tau = 1$. The trend of $\tau \to 1$ is consistent with the fact that multi-head networks with fixed $\tau = 1$ performed as well as networks with trained τ.

Dependence of τ_{net} and τ_{pop} on N: Network-mediated timescales τ_{net} and τ_{pop} generally increase with N in both curricula (Fig. 5, Fig. S4). τ_{net} reflects the contribution of τ and recurrent weights in dynamics of individual neurons. In single-head networks, the mean and variance of τ_{net} follow a similar trend as τ (Fig. 5c), suggesting that changes in τ_{net} can arise from changes in τ. The mean and variance of τ_{net} in multi-head networks increase with N up to some intermediate N, but the pace of increase reduces gradually and saturates for very large Ns (Fig. 5d, top). Given the small τ in multi-head networks, long τ_{net} can only arise from recurrent interactions between neurons. τ_{pop} is the timescale of collective network dynamics (sum of all neurons' activations) and arises from interactions between neurons within the whole network. τ_{pop} exhibits a clear increase with N for both tasks and curricula with comparable values (Fig. S4). These results indicate that in both curricula, collective network dynamics become slower with increasing N, but due to differences in τ, the two curricula employ distinct mechanisms to achieve this.

Dependence of connectivity on N: Multi-head networks have, on average, almost the same total incoming positive and negative weights (with a slight tendency towards larger total negative weights as N increases), leading to relatively balanced dynamics (Fig. 6a). On the other hand, single-head networks have a stronger bias towards more inhibition (negative weights) as N increases. The strong negative weights in single-head networks are required to create stable dynamics in the presence of long single-neuron timescales (Appendix J).

Dimensionality of dynamics: The dimensionality of population activity (measured as the number of principal components that explain 90% of the variance) increases almost linearly with N in the N-parity task but sub-linearly in the N-DMS task, using both curricula, reflecting distinct computational requirements for each task (Fig. 6b, S20, Appendix K). Since computations should be

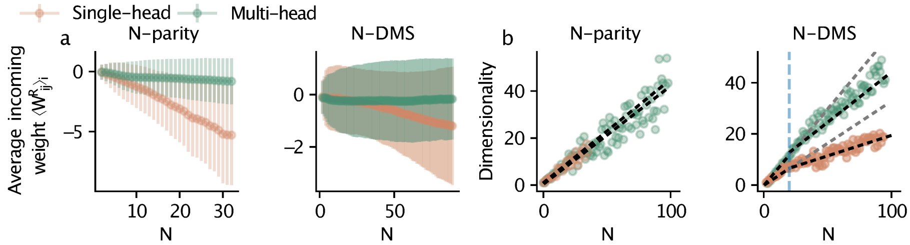


Figure 6: Learned recurrent connectivity and dimensionality of population activity. **a.** The average incoming weight to a neuron remains close to zero for multi-head networks but becomes strongly negative as N increases in single-head networks (example RNN). Error bars - ± STD. **b.** The dimensionality of the activity increases linearly with N in N-parity and sub-linearly in N-DMS task. Dashed lines - linear fit computed for all Ns (N-parity task) and independently for $N \in [0, 20]$ (gray line – extension for visual guidance) and $N \in [20, 100]$ (N-DMS task). Blue line - $N = 20$.

performed at every time step, the increase in dimensionality may be required to map different input patterns (that grow with N) to the same outputs.

Our findings suggest that both curricula give rise to networks with slower and higher dimensional collective dynamics with increasing N, but via distinct mechanisms. In single-head networks, single-neuron properties τ play an important role in creating slow dynamics, which are then stabilized by stronger inhibition in the network. However, in multi-head networks, the slow dynamics should arise from recurrent network interactions. The significant difference in performance of the two curricula suggests that the second mechanism is more effective in solving the task.

4.3 Impact of different curricula on networks robustness

To compare the robustness and retraining capability between the two curricula, we investigate changes in network accuracy resulting from ablations, perturbations, and retraining networks on unseen N. We measure the effects on network performance using a relative accuracy metric with respect to the originally trained network, defined as $\mathrm{acc}_{\mathrm{rel}} := (\mathrm{acc} - 0.5)/(\mathrm{acc}_{\mathrm{base}} - 0.5)$, where $\mathrm{acc}_{\mathrm{base}}$ represents the accuracy of the network before any perturbations or ablations, acc is the measured accuracy after the intervention, and 0.5 is a chance level used for the normalization. If $\mathrm{acc}_{\mathrm{rel}} = 1$, the intervention did not change the accuracy; when $\mathrm{acc}_{\mathrm{rel}} \approx 0$, the intervention reduced the accuracy to chance level. All accuracies are evaluated on the maximal trained N.

Ablation: To examine the relative impact of neurons with different trained τ on network performance, we ablate individual neurons based on their τ and measure the performance without retraining (Appendix L). Specifically, we compare the effect of ablating the 20 longest (4% of the network) and 20 shortest timescale neurons from the network (Fig. 7a,b). For small N and both curricula, ablating individual neurons has only minimal effect (less than 1%) on accuracy. However, for larger N, we observe a considerable difference in the importance of neurons. Single-head networks rely strongly on long-timescale neurons (Fig. 7a), such that ablating them reduces the performance much more than for short-timescale neurons. In contrast, multi-head networks exhibit greater robustness against ablation, and their accuracy is more affected when short-timescale neurons (i.e., neurons with $\tau = 1$) are ablated (Fig. 7b). Note that in single-head networks, the average of the 20 longest single-neuron timescales is 2.7 times longer than in multi-head networks.

Perturbation: We perturb W^R and τ with strength ε as (Wu et al., 2020)

$$\widetilde{W}^R = W^R + \varepsilon \frac{\xi_W}{||\xi_W||}||W^R||, \quad \tilde{\tau} = \tau + \varepsilon\left|\frac{\xi_\tau}{||\xi_\tau||}||\tau||\right|, \quad \xi_W \sim \mathcal{N}(0, \mathbb{I}^{n\times n}), \quad \xi_\tau \sim \mathcal{N}(0, \mathbb{I}^n). \tag{4}$$

$||\cdot||$ represents Frobenius norm and $|\cdot|$ the absolute value. τ is perturbed positively to avoid $\tau < 1$. Multi-head networks are more robust to perturbations (Fig. 7c,d). The robustness to changes in W^R is noteworthy since these networks rely on connectivity to mediate long timescales.

Retraining: We evaluate the performance of networks that solve $N = 16$, when retrained without curriculum (without training on intermediate Ns) for an arbitrary higher N, after 20 epochs. Multi-

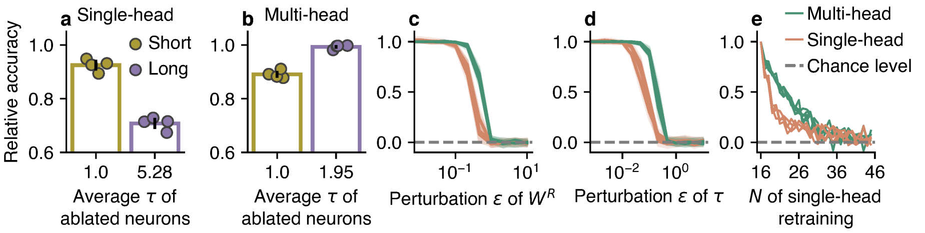


Figure 7: Multi-head networks are more robust to ablation and perturbation, and better retrainable. **(a, b)** Ablating long-timescale neurons largely decreases the performance of single-head networks (a), while multi-head networks (b) are more affected by the ablation of short-timescale neurons ($N = 30$). **(c, d)** Multi-head networks are more robust against perturbations of recurrent connectivity W^R and τ than single-head networks (note the log-scale x-axis, $N = 30$). **(e)** Multi-head networks retrain faster: They achieve higher relative accuracy when retrained for 20 epochs without a curriculum for a higher N. Networks trained to solve $N = 16$ are retrained for 20 epochs to solve higher N without a curriculum to compare re-trainability. Bars - mean; dots and lines - 4 networks for each curriculum; error bars and shades - ± STD.

head networks show a superior retraining ability compared to single-head networks for at least 10 N beyond $N = 16$ (Fig. 7e). This finding suggests that multi-head networks are better at learning the underlying task and adjust faster to a new, larger N even when skipping intermediate Ns.

5 Relation to Neuroscience

Continuous-time setting: So far, we described the tasks and network dynamics in discrete time with time-step $\Delta t = 1$. However, to make the connection to more realistic settings, such as neuroscience, we need to describe the dynamics in continuous time. For this purpose, we consider that each input digit is presented to the network for a certain time T. In neuroscience experiments (e.g., DMS task), T is often set to 250-500 ms (Meyer et al., 2011; Qi et al., 2011; Kim & Sejnowski, 2021).

First, we show that for a fixed T, the discretization time step Δt does not affect the networks' performance and dynamics. We train the networks with each curriculum using a variety of Δt (Appendix B). In these networks, the performance on the test data is comparable for different Δt, even for values much smaller than what was included during the training, mimicking continuous-time dynamics (Fig. S5 c-d). Moreover, similar to our previous results, we find that single-head networks increase their τ with N, while multi-head networks try to reach $\tau \to T$ (Fig. S5 a,b).

Next, we test how changes in input presentation time T affect the dynamics. We set $T = k\Delta t$, where $\Delta t = 1$. We find that for all k, single-head networks increase τ with N, whereas multi-head networks' τ tends to k, thus matching the timescales of the input changes (Appendix C).

Timescales and learning in the biological neural networks: Our findings suggest that networks that are required to solve tasks with larger memory requirements should develop longer timescales. This largely agrees with findings in the brain: higher cortical areas that are involved in cognitive processes with larger memory requirements (e.g., working memory, evidence accumulation) have longer timescales than sensory areas (Murray et al., 2014). Moreover, we show that developing longer network timescales via changes in network connectivity is a superior solution (in terms of performance and stability) than using longer single-neuron timescales. This is consistent with findings that neural timescales in primate visual cortex adapt to task demands via recurrent network interactions rather than biophysical time constants (Zeraati et al., 2023). Moreover, this result aligns with the learning strategy of biological neural networks, which primarily relies on changes in synaptic strengths rather than modifying the biophysical time constants of individual neurons. While such changes do happen in biology (via protein turnover (Sun & Schuman, 2022), calcium currents (Tiganj et al., 2015), and other mechanisms), synaptic strength modification is overwhelmingly the mechanism by which biological networks learn.

6 Related work

Previous works independently investigated the role of neuronal and network-mediated timescales in solving memory tasks and proposed inconsistent solutions. Studies focusing on neuronal aspect suggested heterogeneous and adaptable neuronal properties (e.g., membrane time constant) as an optimal mechanism (Perez-Nieves et al., 2021; Mahto et al., 2021; Smith et al., 2023a; Quax et al., 2020). At the same time, other studies presented that network-mediated mechanisms like balanced dynamics (Lim & Goldman, 2013), strong inhibition (Kim & Sejnowski, 2021) or homeostatic plasticity (Cramer et al., 2020; 2023) can create timescales required for memory tasks. For a single neuron modeled with multiple memory units, long timescales were shown to be instrumental in solving memory tasks (Spieler et al., 2023). Here, we explicitly compare these mechanisms and show that while both can be useful for learning long-memory tasks, applying network-mediated mechanisms leads to faster training and more robust solutions.

We find that the difference between mechanisms is revealed mainly in the context of distinct learning objectives defined by curricula. This is an important distinction with previous work, since the role of timescales has been often studied when RNNs solve a single task (e.g., single-head DMS), without considering learning dynamics or the potential for catastrophic forgetting. We relate the mechanisms of task-dependent timescale with the learning dynamics of RNNs across curricula. The use of curricula in our study is inspired by previous work suggesting curriculum learning as a fitness landscape-smoothing mechanism that can enable the gradual learning of highly complex tasks (Elman, 1993; Bengio et al., 2009; Krueger & Dayan, 2009) and be used to uncover distinct learning mechanisms (Kepple et al., 2022; Dekker et al., 2022). Here, we extend these findings by demonstrating how different curricula can push networks towards adopting different strategies to develop slow collective dynamics required for solving long-memory tasks.

7 Discussion

We find that to solve long-memory tasks, RNNs develop high-dimensional activity with slow timescales via two distinct combinations of connectivity and single-neuron timescales. While single-head networks crucially rely on the long single-neuron timescales to perform the task, multi-head networks prefer a constant single-neuron timescale and solve the task relying only on the long timescales emerging from recurrent interactions. We show that developing long timescales via recurrent interactions instead of single-neuron properties is optimal for learning memory tasks and leads to more stable and robust solutions, which can be a beneficial strategy for brain computations.

Our findings suggest that training networks on *sets* of related memory tasks instead of a single task improves performance and robustness. By progressively shaping the loss function with a curriculum to include performance evaluations on sub-tasks that are known to correlate with the desired task, we can smooth the loss landscape of our network to allow training for difficult tasks that were previously unsolvable. In this way, choosing an appropriate curriculum can act as a powerful regularization.

Limitations: Our study considers two relatively simple tasks with explicitly controllable memory requirements. In follow-up studies, it would be important to test our observations in more sophisticated tasks and investigate whether our results apply to other architectures and optimizers. Additionally, our approach is suitable only for a set of tasks with controllably increasing memory requirements, where the different versions of the same task can be simultaneously performed on the same data (multi-head training). This is a relatively strong constraint, and future research expanding our findings could focus on generalizing the multi-head curriculum for training more realistic tasks. Time series reconstruction is a potential task that can be used to uncover generative dynamical systems from data (Durstewitz et al., 2023). We proposed a potential experiment in Appendix D.

Our current model is a crude approximation of biological neural networks, and more plausible architectures (spiking models, distinct neuron types) could be studied. Finally, biological neural networks can produce long timescales via various other mechanisms we did not consider here (short-term plasticity (Hu et al., 2021), adaptation (Salaj et al., 2021; Beiran & Ostojic, 2019), synaptic delays, etc). A follow-up study could investigate whether our findings extend to more plausible networks incorporating such additional mechanisms.

Acknowledgments

This work was supported by a Sofja Kovalevskaja Award from the Alexander von Humboldt Foundation, endowed by the Federal Ministry of Education and Research (SK, RZ, EG, AL), the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC number 2064/1 - Project number 390727645 (RZ, EG), and Else Kröner Medical Scientist Kolleg "ClinbrAIn: Artificial Intelligence for Clinical Brain Research" (TJS). We acknowledge the support from the BMBF through the Tübingen AI Center (FKZ: 01IS18039B), International Max Planck Research School for the Mechanisms of Mental Function and Dysfunction (IMPRS-MMFD), and International Max Planck Research School for Intelligent Systems (IMPRS-IS). We thank Victor Buendía for valuable discussions.

References

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974. doi: 10.1109/TAC.1974.1100705. URL https://ieeexplore.ieee.org/document/1100705.

Brice Bathellier, Derek L. Buhl, Riccardo Accolla, and Alan Carleton. Dynamic ensemble odor coding in the mammalian olfactory bulb: Sensory information at different timescales. *Neuron*, 57(4):586–598, 2008. ISSN 0896-6273. doi: https://doi.org/10.1016/j.neuron.2008.02.011. URL https://www.sciencedirect.com/science/article/pii/S0896627308001347.

Manuel Beiran and Srdjan Ostojic. Contrasting the effects of adaptation and synaptic filtering on the timescales of dynamics in recurrent networks. *PLOS Computational Biology*, 15(3):e1006893, March 2019. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1006893. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1006893.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 41–48, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553380. URL https://doi.org/10.1145/1553374.1553380.

Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. *arXiv preprint arXiv:1206.6392*, 2012. URL https://arxiv.org/abs/1206.6392.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015. URL https://arxiv.org/abs/1511.06349.

Sean E. Cavanagh, Laurence T. Hunt, and Steven W. Kennerley. A Diversity of Intrinsic Timescales Underlie Neural Computations. *Frontiers in Neural Circuits*, 14, 2020. ISSN 1662-5110. doi: 10.3389/fncir.2020.615626. URL https://www.frontiersin.org/articles/10.3389/fncir.2020.615626/full?field=&id=615626&journalName=Frontiers_in_Neural_Circuits. Publisher: Frontiers.

Rishidev Chaudhuri, Alberto Bernacchia, and Xiao-Jing Wang. A diversity of localized timescales in network activity. *eLife*, 3, January 2014. ISSN 2050-084X. doi: 10.7554/eLife.01239. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3895880/.

Rishidev Chaudhuri, Kenneth Knoblauch, Marie-Alice Gariel, Henry Kennedy, and Xiao-Jing Wang. A Large-Scale Circuit Mechanism for Hierarchical Dynamical Processing in the Primate Cortex. *Neuron*, 88(2):419–431, October 2015. ISSN 0896-6273. doi: 10.1016/j.neuron.2015.09.008. URL http://www.sciencedirect.com/science/article/pii/S0896627315007655.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. URL `https://arxiv.org/abs/1412.3555`.

Benjamin Cramer, David Stöckel, Markus Kreft, Michael Wibral, Johannes Schemmel, Karlheinz Meier, and Viola Priesemann. Control of criticality and computation in spiking neuromorphic networks with plasticity. *Nature communications*, 11(1):2853, 2020. URL `https://www.nature.com/articles/s41467-020-16548-3`.

Benjamin Cramer, Markus Kreft, Sebastian Billaudelle, Vitali Karasenko, Aron Leibfried, Eric Müller, Philipp Spilger, Johannes Weis, Johannes Schemmel, Miguel A Muñoz, et al. Autocorrelations from emergent bistability in homeostatic spiking neural networks on neuromorphic hardware. *Physical Review Research*, 5(3):033035, 2023. URL `https://journals.aps.org/prresearch/abstract/10.1103/PhysRevResearch.5.033035`.

Ronald B Dekker, Fabian Otto, and Christopher Summerfield. Curriculum learning for human compositional generalization. *Proceedings of the National Academy of Sciences*, 119(41):e2205582119, 2022. URL `https://www.pnas.org/doi/10.1073/pnas.2205582119`.

Renato Duarte, Alexander Seeholzer, Karl Zilles, and Abigail Morrison. Synaptic patterning and the timescales of cortical dynamics. *Current Opinion in Neurobiology*, 43:156–165, April 2017. ISSN 0959-4388. doi: 10.1016/j.conb.2017.02.007. URL `https://www.sciencedirect.com/science/article/pii/S0959438817300545`.

Daniel Durstewitz, Georgia Koppe, and Max Ingo Thurm. Reconstructing computational system dynamics from neural data with recurrent neural networks. *Nature Reviews Neuroscience*, pp. 1–18, 2023. URL `https://www.nature.com/articles/s41583-023-00740-7`.

Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990. URL `https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1402_1`.

Jeffrey L Elman. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99, 1993. URL `https://www.sciencedirect.com/science/article/pii/0010027793900584`.

Wei Fang, Zhaofei Yu, Yanqi Chen, Timothee Masquelier, Tiejun Huang, and Yonghong Tian. Incorporating Learnable Membrane Time Constant to Enhance Learning of Spiking Neural Networks, August 2021. URL `http://arxiv.org/abs/2007.05785`. arXiv:2007.05785 [cs].

Richard Gao, Ruud L van den Brink, Thomas Pfeffer, and Bradley Voytek. Neuronal timescales are functionally dynamic and shaped by cortical microarchitecture. *eLife*, 9:e61277, November 2020. ISSN 2050-084X. doi: 10.7554/eLife.61277. URL `https://doi.org/10.7554/eLife.61277`. Publisher: eLife Sciences Publications, Ltd.

Julijana Gjorgjieva, Guillaume Drion, and Eve Marder. Computational implications of biophysical diversity and multiple timescales in neurons and synapses for circuit performance. *Current Opinion in Neurobiology*, 37:44–52, April 2016. ISSN 0959-4388. doi: 10.1016/j.conb.2015.12.008. URL `https://www.sciencedirect.com/science/article/pii/S0959438815001865`.

Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013. URL `https://arxiv.org/abs/1308.0850`.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pp. 6645–6649. Ieee, 2013. URL `https://arxiv.org/abs/1303.5778`.

David Ha and Douglas Eck. A neural representation of sketch drawings. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=Hy6GHpkCW`.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997. URL https://direct.mit.edu/neco/article-abstract/9/8/1735/6109/Long-Short-Term-Memory?redirectedFrom=fulltext.

Brian Hu, Marina E Garrett, Peter A Groblewski, Douglas R Ollerenshaw, Jiaqi Shang, Kate Roll, Sahar Manavi, Christof Koch, Shawn R Olsen, and Stefan Mihalas. Adaptation supports short-term memory in a visual change detection task. *PLoS computational biology*, 17 (9):e1009246, 2021. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1009246.

Shailee Jain, Vy Vo, Shivangi Mahto, Amanda LeBel, Javier S Turek, and Alexander Huth. Interpretable multi-timescale models for predicting fMRI responses to continuous natural speech. *Adv. Neural Inf. Process. Syst.*, 33:13738–13749, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/9e9a30b74c49d07d8150c8c83b1ccf07-Abstract.html.

John Jonides, Richard L. Lewis, Derek Evan Nee, Cindy A. Lustig, Marc G. Berman, and Katherine Sledge Moore. The Mind and Brain of Short-Term Memory. *Annual Review of Psychology*, 59(1):193–224, 2008. doi: 10.1146/annurev.psych.59.103006.093615. URL https://doi.org/10.1146/annurev.psych.59.103006.093615.

Daniel R. Kepple, Rainer Engelken, and Kanaka Rajan. Curriculum learning as a tool to uncover learning principles in the brain. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TpJMvo0_pu-.

Robert Kim and Terrence J. Sejnowski. Strong inhibitory signaling underlies stable temporal dynamics and working memory in spiking neural networks. *Nature Neuroscience*, 24(1):129–139, January 2021. ISSN 1546-1726. doi: 10.1038/s41593-020-00753-w. URL https://www.nature.com/articles/s41593-020-00753-w.

Kai A. Krueger and Peter Dayan. Flexible shaping: How learning in small steps helps. *Cognition*, 110(3):380–394, 2009. ISSN 0010-0277. doi: https://doi.org/10.1016/j.cognition.2008.11.014. URL https://www.sciencedirect.com/science/article/pii/S0010027708002850.

Sukbin Lim and Mark Goldman. Balanced cortical microcircuitry for maintaining information in working memory. *Nature neuroscience*, 16, 2013. doi: 10.1038/nn.3492. URL https://www.nature.com/articles/nn.3492.

Zachary C Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015. URL https://arxiv.org/abs/1506.00019.

Ashok Litwin-Kumar and Brent Doiron. Slow dynamics and high variability in balanced cortical networks with clustered connections. *Nature Neuroscience*, 15(11):1498–1505, November 2012. ISSN 1546-1726. doi: 10.1038/nn.3220. URL https://www.nature.com/articles/nn.3220. Number: 11 Publisher: Nature Publishing Group.

Shivangi Mahto, Vy Ai Vo, Javier S. Turek, and Alexander Huth. Multi-timescale representation learning in {lstm} language models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=9ITXiTrAoT.

Travis Meyer, Xue-Lian Qi, Terrence R Stanford, and Christos Constantinidis. Stimulus selectivity in dorsal and ventral prefrontal cortex after training in working memory tasks. *Journal of neuroscience*, 31(17):6266–6276, 2011. URL https://www.jneurosci.org/content/31/17/6266.

John D. Murray, Alberto Bernacchia, David J. Freedman, Ranulfo Romo, Jonathan D. Wallis, Xinying Cai, Camillo Padoa-Schioppa, Tatiana Pasternak, Hyojung Seo, Daeyeol Lee, and Xiao-Jing Wang. A hierarchy of intrinsic timescales across primate cortex. *Nature Neuroscience*, 17(12):1661–1663, December 2014. ISSN 1546-1726. doi: 10.1038/nn.3862. URL https://www.nature.com/articles/nn.3862.

Srdjan Ostojic. Two types of asynchronous activity in networks of excitatory and inhibitory spiking neurons. *Nature Neuroscience*, 17(4):594–600, April 2014. ISSN 1546-1726. doi: 10.1038/nn.3658. URL https://www.nature.com/articles/nn.3658.

Stefano Panzeri, Nicolas Brunel, Nikos K. Logothetis, and Christoph Kayser. Sensory neural codes using multiplexed temporal scales. *Trends in Neurosciences*, 33(3):111–120, March 2010. ISSN 0166-2236. doi: 10.1016/j.tins.2009.12.001. URL http://www.sciencedirect.com/science/article/pii/S0166223609002008.

Pranay Pasula. Real world time series benchmark datasets with distribution shifts: Global crude oil price and volatility. *arXiv preprint arXiv:2308.10846*, 2023. URL https://arxiv.org/abs/2308.10846.

Nicolas Perez-Nieves, Vincent C. H. Leung, Pier Luigi Dragotti, and Dan F. M. Goodman. Neural heterogeneity promotes robust learning. *Nature Communications*, 12(1):5791, October 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-26022-3. URL https://www.nature.com/articles/s41467-021-26022-3.

Xue-Lian Qi, Travis Meyer, Terrence R Stanford, and Christos Constantinidis. Changes in prefrontal neuronal activity after learning to perform a spatial working memory task. *Cerebral cortex*, 21(12):2722–2732, 2011. URL https://academic.oup.com/cercor/article/21/12/2722/295413.

Silvan C. Quax, Michele D'Asaro, and Marcel A. J. van Gerven. Adaptive time scales in recurrent neural networks. *Scientific Reports*, 10(1):11360, July 2020. ISSN 2045-2322. doi: 10.1038/s41598-020-68169-x. URL https://www.nature.com/articles/s41598-020-68169-x. Number: 1 Publisher: Nature Publishing Group.

Shervin Safavi, Matthew Chalk, Nikos Logothetis, and Anna Levina. Signatures of criticality in efficient coding networks. *bioRxiv*, pp. 2023–02, 2023. URL https://www.biorxiv.org/content/10.1101/2023.02.14.528465v1.

Darjan Salaj, Anand Subramoney, Ceca Kraisnikovic, Guillaume Bellec, Robert Legenstein, and Wolfgang Maass. Spike frequency adaptation supports network computations on temporally dispersed information. *Elife*, 10:e65459, 2021. URL https://elifesciences.org/articles/65459.

Yan-Liang Shi, Roxana Zeraati, Anna Levina, and Tatiana A Engel. Spatial and temporal correlations in neural networks with structured connectivity. *Physical Review Research*, 5 (1):013005, 2023. URL https://journals.aps.org/prresearch/abstract/10.1103/PhysRevResearch.5.013005.

Jimmy T.H. Smith, Andrew Warrington, and Scott Linderman. Simplified state space layers for sequence modeling. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=Ai8Hw3AXqks.

Jimmy T.H. Smith, Andrew Warrington, and Scott Linderman. Simplified state space layers for sequence modeling. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=Ai8Hw3AXqks.

Aaron Spieler, Nasim Rahaman, Georg Martius, Bernhard Schölkopf, and Anna Levina. The elm neuron: an efficient and expressive cortical neuron model can solve long-horizon tasks. *arXiv preprint arXiv:2306.16922*, 2023. URL https://arxiv.org/abs/2306.16922.

Chao Sun and Erin M. Schuman. Logistics of neuronal protein turnover: Numbers and mechanisms. *Molecular and Cellular Neuroscience*, 123:103793, 2022. ISSN 1044-7431. doi: https://doi.org/10.1016/j.mcn.2022.103793. URL https://www.sciencedirect.com/science/article/pii/S1044743122000999.

Corentin Tallec and Yann Ollivier. Can recurrent neural networks warp time? *arXiv preprint arXiv:1804.11188*, 2018. URL https://arxiv.org/abs/1804.11188.

Zoran Tiganj, Michael E. Hasselmo, and Marc W. Howard. A simple biophysically plausible model for long time constants in single neurons. *Hippocampus*, 25(1):27–37, 2015. doi: https://doi.org/10.1002/hipo.22347. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/hipo.22347.

José F Torres, Dalil Hadjout, Abderrazak Sebaa, Francisco Martínez-Álvarez, and Alicia Troncoso. Deep learning for time series forecasting: a survey. *Big Data*, 9(1):3–21, 2021. URL https://www.liebertpub.com/doi/10.1089/big.2020.0159.

Alexander van Meegen and Sacha J. van Albada. Microscopic theory of intrinsic timescales in spiking neural networks. *Physical Review Research*, 3(4):043077, October 2021. doi: 10.1103/PhysRevResearch.3.043077. URL https://link.aps.org/doi/10.1103/PhysRevResearch.3.043077.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33:2958–2969, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1ef91c212e30e14bf125e9374262401f-Abstract.html.

Bojian Yin, Federico Corradi, and Sander M. Bohté. Effective and Efficient Computation with Multiple-timescale Spiking Recurrent Neural Networks. In *International Conference on Neuromorphic Systems 2020*, ICONS 2020, pp. 1–8, New York, NY, USA, July 2020. Association for Computing Machinery. ISBN 978-1-4503-8851-1. doi: 10.1145/3407197.3407225. URL https://dl.acm.org/doi/10.1145/3407197.3407225.

Yong Yu, Xiaosheng Si, Changhua Hu, and Jianxun Zhang. A review of recurrent neural networks: Lstm cells and network architectures. *Neural computation*, 31(7):1235–1270, 2019. URL https://ieeexplore.ieee.org/document/8737887.

Roxana Zeraati, Tatiana A. Engel, and Anna Levina. A flexible Bayesian framework for unbiased estimation of timescales. *Nature Computational Science*, 2(3):193–204, March 2022. ISSN 2662-8457. doi: 10.1038/s43588-022-00214-3. URL https://www.nature.com/articles/s43588-022-00214-3.

Roxana Zeraati, Yan-Liang Shi, Nicholas A Steinmetz, Marc A Gieselmann, Alexander Thiele, Tirin Moore, Anna Levina, and Tatiana A Engel. Intrinsic timescales in the visual cortex change with selective attention and reflect spatial connectivity. *Nature Communications*, 14(1):1858, 2023. URL https://www.nature.com/articles/s41467-023-37613-7.

Appendix

A Different types and locations of nonlinearity

In order to verify that our results are robust with respect to the type of nonlinearity used in the network, we train RNNs using two of the most commonly used nonlinearities: ReLU and Tanh. We find that in both cases, the training performance is similar to leaky ReLU, and the development of single-neuron and network-mediated timescales follow the same trajectory as N increases (Fig. S7).

In some implementations of leaky-RNN, the neural self-interaction is linear and located outside of the nonlinearity (cf. equ.1)

$$r_i(t) = \left(1 - \frac{\Delta t}{\tau_i}\right) \cdot r_i(t-\Delta t) + \left[\frac{\Delta t}{\tau_i} \cdot \left(\sum_{i \neq j} W_{ij}^R \cdot r_j(t-\Delta t) + W_i^I \cdot S(t) + b^R + b^I\right)\right]_\alpha . \quad (5)$$

with the explicit time discretization Δt. The input is presented for time duration $T = k\Delta t$ with input-update time steps k. In the main text, we chose $k = 1$ and $\Delta t = 1$. We discuss $k > 1$ in Appendix C and $\Delta t < 1$ in Appendix B.

We verify that training RNNs with this implementation gives similar training dynamics and trajectories of τ and τ_{net} with increasing N (Fig. S8), for both curricula. Furthermore, we find that, for large N, ablating neurons with long τ in single-head networks and neurons with short τ in multi-head networks reduces the performance significantly, compatible with the findings in the main text (Fig. S9, cf. Fig. 7). We also verify that the performance of the model depends on the initialization of τ and its trainability in the same way regardless of the location of the nonlinearity (Fig. S11, cf. Fig. 4).

B Changing time discretization

In computational neuroscience, the single neuron dynamics are typically captured by the differential equations that need to be discretized for running numerical simulations and training networks. However, the discretization can be important for stability and internal representation of the model and the task. In the main text, we used Eq. 5 with $\Delta t = T = 1$. For simplicity of notation, we take in the rest of this section $T = 1$. We train networks with different values of Δt (a different Δt for each training batch), so they can perform the same task independent time discretization. We take $\Delta t = 1/n$ with $n \in \mathbb{N}$ and train the network while keeping the duration of each stimulus presentation in units of time constant (which means that with larger n, it would be presented for more time steps). The flexible framework for time discretization allows us to train with multiple Δt simultaneously. Then, we test whether the network can solve the same task but with Δt not included in their training set.

We find that in networks trained with multiple Δt, the single-neuron timescales τ follow a similar trajectory as the results in the main text, independent of Δt (Fig. S5a,b compared to Fig. 4b). Multi-head networks adjust their τ to converge to $n\Delta t = 1$, while single-head networks increase their individual neuron timescale. Moreover, the networks can generalize (without retraining) the task to smaller Δt than what was included in their training set, in single- and multi-head networks. Interestingly, the performance decreases slowly when Δt becomes smaller than the training set, but abruptly when it becomes larger (Fig. S5c,d). The performance is best when training with multiple Δt, but qualitatively, the result is similar for a single, small enough Δt (Fig. S5e).

C Changing the duration of the input presentation

In our tasks, the input contains two timescales. First is the duration of presentation of each input digit $T = k \cdot T_{\min}$, with $T_{\min}$ a minimal considered duration of stimulus presentation measured in milliseconds. Second is the timescale of the task's memory N. In the main text, we consider the situation of $k = 1$, but in general, k acts as a time-rescaling parameter and defines one unit of time for the task performance. Here, we train the RNNs with different values of $k \in \{2, 3, 5, 10\}$ and check the trajectories of changing τ with N depending on k. We find that similar to the case with $k = 1$, single-head networks trained with $k > 1$ increase their τ with N, while multi-head networks try to keep τ close to k (Fig. S6). Moreover, tasks with $k > 1$ are generally more difficult to solve since the input needs to be tracked over $N \cdot k$ time steps. Hence, as k grows, RNNs would reach smaller N within the same number of training epochs. The changes in values of τ after rescaling with k might be due to nonlinear interactions in the network arising from the combination of different N and k.

D Proposed additional task: Temporal pattern generation

For future research, the task variety can be extended to include the temporal pattern generation, which is a continuous-time task that is often used to evaluate RNNs (Durstewitz et al., 2023). The classic variation of the task involved an RNN receiving either random noise or no input and having to produce a target time series as output (usually a sum of sine waves with different frequencies).

A variation of the task we could consider for testing our model is the following:

Single-head: On the first step of the curriculum, we train the network to produce a single sine wave with frequency f_1, setting the target sequence to be $y_{N=1} = \sin(2\pi \cdot f_1 \cdot t)$.

Then, for each curriculum step, we complexify the target sequence by setting the new target as:

$$y_{N=m} = \sum_{i=1}^{m} \sin(2\pi \cdot f_i \cdot t), \tag{6}$$

for $f_1 > f_2 > \cdots > f_m$. In this way, as the newly added frequencies decrease, a need arises for the network to develop longer timescales.

Multi-head: Unlike the single-head network where the RNN needs to produce only one target time series $y_{N=m}$ at the m-th step of the curriculum, in the multi-head curriculum, the network produces m output time series $Y = \{y_{N=1}, \ldots, y_{N=m}\}$.

E Effects of training without a curriculum on the N-DMS task

We investigate the negative effects of not using a curriculum during training for the N-DMS task to extend our results from Fig. 3a. We show in Fig. S13 that similar to the N-Parity task, networks rapidly lose the ability to solve the N-DMS task as N increases when training without a curriculum. Interestingly, the two tasks differ in the way they fail to be solved despite using identical optimizers. In all of our results, the N-DMS task tends to be easier to solve for larger N. However, despite the relative success these networks have with the N-DMS task, their drop-off in training these networks is much steeper when comparing the curves from Fig. 3a and Fig. S13. In Fig. S13, tasks $N < 15$ get solved in only 1 or 2 epochs, however between $15 < N < 20$ the networks rapidly slow down in their ability to train until completely failing for $N > 20$ even when given longer training time. We can infer from these results that different tasks have varying degrees to which they benefit from a particular curriculum.

F Intermediate curricula: multi-head with a sliding window

The two curricula discussed in the main text (single-head and multi-head) represent two extreme cases. In the single-head curriculum, at each step of the curriculum, RNNs are trained to solve a new N without requiring to remember the solution to the previous Ns. On the other hand, in the multi-head curriculum, RNNs need to remember the solution to all the previous Ns in addition to the new N. Here, we test the behavior of curricula that lie in between the two extreme cases.

The intermediate curricula involve the simultaneous training of multiple heads, similar to the multi-head curriculum, but instead of adding new heads at each curriculum step, we train a fixed number of heads and only shift the Ns, which they are trained for according to a sliding window. We consider the number of heads to be 10, and start the training for $N \in [2, \ldots, 11]$. In the next steps of the curriculum, we use the already trained network to initialize another network which we train for $N + w$ (e.g., $N \in [2 + w, \ldots, 11 + w]$), where $w \in \{1, 3, 5\}$ indicates the size of the sliding window. For each w, we train 4 different networks (i.e., 4 different initialization). For the following analyses, we trained the networks on the N-parity task.

We find that networks trained with the multi-head-sliding curriculum generally demonstrate an in-between behavior compared with the extreme curricula, but the results also depend on the size of the sliding window. Within 1000 training epochs, the maximal N these networks can solve (with $> 98\%$ accuracy) is in between the maximal N of single- and multi-head curricula, depending on the sliding window. Networks with a larger sliding window can solve a higher maximal N, indicating that a large sliding window not only does not slow down the training but also provides a more efficient curriculum to learn higher Ns (Fig. S12a). Moreover, in multi-head-sliding networks, single-neuron (τ) and network-mediated (τ_{net}) timescales have values in between single-head and multi-head curricula (Fig. S12b). However, both τ and τ_{net} grow with N similar to single-head networks, with the pace of growth reducing for larger sliding windows.

Similar to the main text (Fig. 7c,d,e), we perform the perturbation and retraining analysis on multi-head-sliding networks trained with $w = 5$. The relative accuracy after perturbation of recurrent weights W^R and timescales τ for these networks lies between the two extremes (Fig. S14a, b). However, the retraining analysis suggests that multi-head-sliding networks can be retrained better for higher new Ns (Fig. S14c,d). If the network is originally retrained for a small N (e.g., $N = 16$), the retraining relative accuracy is similar between multi-head and sliding networks but is larger than single-head networks. For networks trained for larger Ns (e.g., $N = 31$), sliding networks exhibit a superior retraining ability compared to the other two curricula. These results suggest that the curriculum with the sliding window helps multi-head networks to better adjust to new Ns.

G Single- and multi-head curricula for training GRU and LSTM

The results presented in the main text were generated using a modified version of a vanilla RNN (leaky-RNN) with an explicit definition of the timescale parameter τ. To test whether the difficulties in training for long memory tasks without curriculum would carry over to recurrent networks that were specifically designed for long memory tasks, we train two other architectures, an LSTM (long short-term memory) and a GRU (gated recurrent unit) on the N-parity task for increasing N, with and without a curriculum. Both the GRU and LSTM have similar network sizes to the RNN with 500 neurons, though they differ in their activation functions (the RNN used a single leakyReLU whereas the GRU/LSTMs have both sigmoids and tanhs for different gates). Furthermore, in contrast with the RNNs, an Adam optimizer is used with learning rate $lr = 10^{-3}$ and the input signals to the models take values $\in \{-1, 1\}$ (to have a zero-mean input signal).

We find that for both architectures, training the networks without a curriculum is extremely slow for large N and relatively unstable for small N and probably requires strict hyper-parameter tuning (Fig. S15a). Without additional hyper-parameter tuning, introducing the multi-head curriculum speeds up the training significantly, and both architectures can easily learn the N-parity task with large N similar to the leaky-RNN (Fig. S15b). Moreover, similar to RNNs, the multi-head curriculum has a higher training speed than the single-head curriculum (Fig. S16). Our results indicate that GRUs and LSTMs are subject to similar training dynamics as RNNs used in the main text and the multi-head curriculum is an optimal curriculum regardless of the RNN architecture. The advantage of using the leaky-RNN architecture is that its parameters are easier to interpret, and it allows us to study better the mechanisms underlying each curriculum by explicitly studying the role of timescales.

H Backward and forward retraining of networks

To understand how trained models develop their ability to create longer timescales throughout the curriculum as well as their backward compatibility and robustness to catastrophic forgetting, we measure the retrainability of models trained on a task with memory N on a different task with memory N^*. We freeze all parameters of a trained network except the final readout layer weights which are retrained on an N^* task. Specifically, we load models trained for $N \in [2, \ldots, 19]$ and retrain them on a new $N^* \in [2, \ldots, N{+}2]$, independently for each N^*, for a maximum of 10 epochs or until its accuracy was above 98%. Note that we retrain both single- and multi-head networks as single-head.

We find that the multi-head networks exhibit near-perfect backward compatibility as well as better forward compatibility than the single-head models (Fig. S17), while single-head networks suffer from catastrophic forgetting. For the multi-head networks, the backward compatibility is enforced through the loss function (as is the case in the multi-head curriculum) hence, the necessary representations for $N^* < N$ persist. However, the multi-head curriculum also has positive implications for forward compatibility, which is evident in the off-diagonal entries of the accuracy where $N^* > N$ (to the right of the dotted line) when compared to the single-head values.

I Emergence of curriculum during multi-head training

In the multi-head curriculum, the difficulty of the task increases gradually; a new head with a larger N is added at each step of the curriculum. In the main text, we discussed that networks trained with such a curriculum generally train well up to large Ns. Here we ask whether this optimal curriculum can emerge by itself if we train a network with multiple heads, but without any predefined curricula. For this analysis, we train RNNs with 19 ($N \in [2, \ldots, 20]$) and 39 heads ($N \in [2, \ldots, 40]$) to solve all the available Ns simultaneously.

We find that despite the absence of an explicit curriculum, these networks learn the task by generating an internal multi-head curriculum. While all the heads contribute equally to the loss, heads with a smaller N reach the higher accuracy faster (Fig. S18a). However, the speed of training strongly depends on the total number of heads in each network. For the same N, the network with 19 heads reaches the 98% accuracy faster than the network with 39 heads (Fig. S18b), but both networks have a slower training speed when compared to the multi-head curriculum. These results suggest that the multi-head curriculum is an optimal curriculum that can arise naturally during multi-head training and can increase the training speed when applied explicitly.

J Role of strong inhibitory connectivity in single-head networks

The main difference between single- and multi-head networks in terms of connectivity is the stronger inhibitory (negative) connectivity for large N in the single-head networks compared with the relatively balanced connectivity in multi-head networks (Fig. 6a). We hypothesized that larger inhibition in single-head networks is required to keep the dynamics stable in the presence of slow single-neuron timescales τ. To test this hypothesis, we perturb only the inhibitory connections in networks trained with both curricula as:

$$W_{ij}^R = W_{ij} + c \cdot W_{ij}, \qquad \forall\, W_{ij}^R < 0, \tag{7}$$

for a given amount of $c \in [-0.1, 0.1]$. We observe that by reducing the amount of inhibition in single-head networks, the network activity explodes even before reaching the balanced point, i.e., the point when the average incoming weight of neurons becomes 0 (Fig. S19a). On the contrary, multi-head networks are significantly more robust to such perturbations and their activity remains within a reasonable range for a broad range of inhibitory scaling (Fig. S19b).

This difference is most likely attributed to the difference in single-neuron timescales τ between single- and multi-head networks. The single-head networks have a larger average τ compared to the multi-head networks whose average $\tau \approx 1$ for large N. Longer τ leads to neurons with self-sustaining activity, and thus, a stronger inhibition might be required to prevent the runaway activation. Such a relationship can be observed when comparing the average τ and inhibitory strength across networks: for single-head networks as τ grows, the average weight becomes more negative (inhibitory)(Fig. S19c), but such correlation does not exist in multi-head networks (Fig. S19d).

K Dependence of dimensionality of population activity on N

We measure the dimensionality as the number of principal components that explain 90% of the population activity variance. The dimensionality increases with N for both tasks and curricula, but the increase follows a linear relation with N for N-parity task but a sub; linear relation for the N-DMS task (Fig. 6b).

To demonstrate this difference, we fit two separate lines for the data up to $N = 20$ and from $N = 20$ up to the largest N. We observe that for the N-parity task, the slope of two lines largely overlaps, indicating a linear relation. However, for the N-DMS task, the second line clearly has a smaller slope than the first one, indicating a sub-linear growth with N (Fig. S20).

L Ablation details

To test whether neurons with fast or slow timescales (τ) are necessary for computations in the trained RNNs we perform the ablation analysis. For this analysis, we compute the relative accuracy of the model (Eq. 4 in the main text) after removing a single neuron. We ablate neuron i by setting all incoming and outgoing associated weights to zero

$$\begin{aligned} &W^R_{ij}, W^R_{ji} = 0 \qquad \forall j \\ &W^O_i, W^I_i = 0 \end{aligned} \tag{8}$$

Here W_{ij} refers to recurrent weights, W^O_i to input weights and W^O_i to readout weights. To measure the relative accuracy, we simulate the RNN forward using random binary inputs for 1000 time steps after 100 time steps of a burn-in period (to reach the stationary state). Then, we evaluate the accuracy of the network at each time step. We repeat this procedure over 10 trials and compute the average and standard deviation of the relative accuracies across trials.

M Significance of the responses to perturbations of weights and retraining

We investigate the significance of differences between single- and multi-head networks presented in Fig. 7 using a t-test (two-sided, unpaired). Perturbations are computed 10 times for 4 networks per group with results being pooled across networks. Retraining accuracy is computed once per network. Table 1,2, and 3 indicates with stars the significance levels corresponding to p-values below $5e{-}2, 1e{-}2, 1e{-}3, 1e{-}4$, and $1e{-}5$.

Weight Perturbation Strength	p-value	Significance
1.0e-02	8.8e-03	**
2.2e-02	4.5e-01	n/s
4.6e-02	2.9e-01	n/s
1.0e-01	5.1e-15	*****
2.2e-01	2.2e-39	*****
4.6e-01	8.3e-47	*****
1.0e+00	8.6e-04	***
2.2e+00	7.6e-01	n/s
4.6e+00	6.5e-01	n/s
1.0e+01	9.0e-01	n/s

Table 1: Significance of the weights' perturbation for different perturbation sizes Fig. 7c. Two-sided and unpaired t-test, stars indicate p-values below $5e{-}2, 1e{-}2, 1e{-}3, 1e{-}4$, and $1e{-}5$.

Perturbation of τ	**p-value**	**Significance**
1.0e-03	4.3e-01	n/s
2.2e-03	8.9e-01	n/s
4.6e-03	5.4e-01	n/s
1.0e-02	1.2e-02	*
2.2e-02	2.3e-14	*****
4.6e-02	9.1e-29	*****
1.0e-01	1.3e-50	*****
2.2e-01	4.1e-35	*****
4.6e-01	4.8e-01	n/s
1.0e+00	6.0e-01	n/s
2.2e+00	1.3e-01	n/s

Table 2: Significance of the τ's perturbation for different perturbation sizes Fig. 7d. Two-sided and unpaired t-test, stars indicate p-values below $5e{-}2, 1e{-}2, 1e{-}3, 1e{-}4$, and $1e{-}5$.

retraining for N	**p-value**	**Significance**
17	1.7e-03	**
18	9.7e-04	***
19	1.2e-07	*****
20	3.2e-05	****
21	1.7e-05	****
22	1.6e-05	****
23	7.4e-06	*****
24	6.4e-05	****
25	3.1e-04	***
26	1.7e-03	**
27	1.7e-02	*
28	2.1e-03	**
29	7.5e-03	**
30	1.6e-01	n/s
31	1.5e-01	n/s
32	6.5e-01	n/s

Table 3: Significance of the retraining differences between single and multi-head, Fig. 7e. Two-sided and unpaired t-test, stars indicate p-values below $5e{-}2, 1e{-}2, 1e{-}3, 1e{-}4$, and $1e{-}5$.

N CODE AND DATA AVAILABILITY

Codes for training and evaluating the RNNs and reproducing the experiments (e.g., measuring timescales, performing ablations, etc.) together with example trained networks are available on GitHub at https://github.com/LevinaLab/rnn_timescale_public (more details in README).

O Supplementary figures

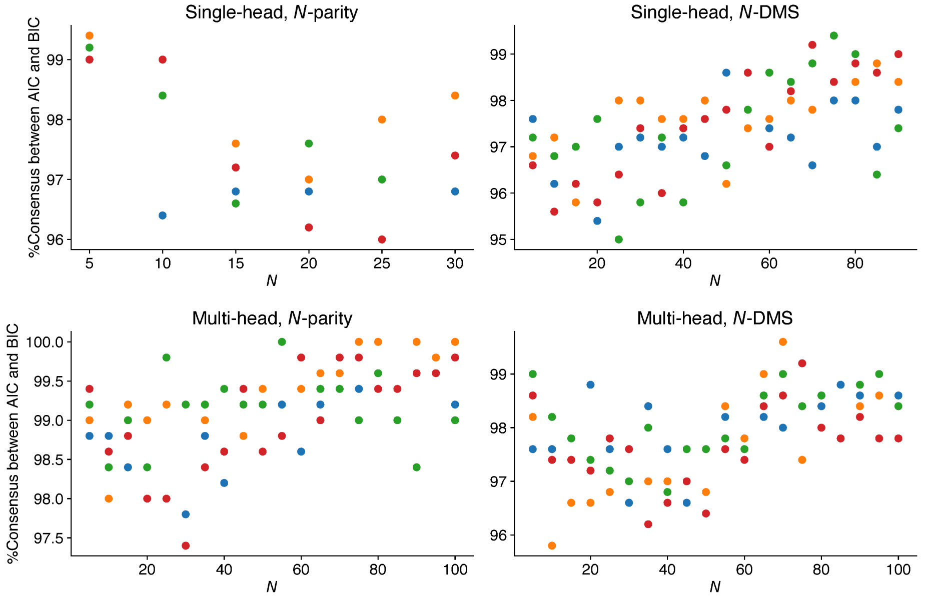


Figure S1: AIC and BIC choose the same model for most neurons. We fitted AC of each neuron's activity with single- and double-exponential functions and used AIC or BIC to select the best-fitting models. The results show that for above 95% of neurons, the two criteria select the same model. The colors of the dots indicate different networks (4 networks for each task, curriculum and N).

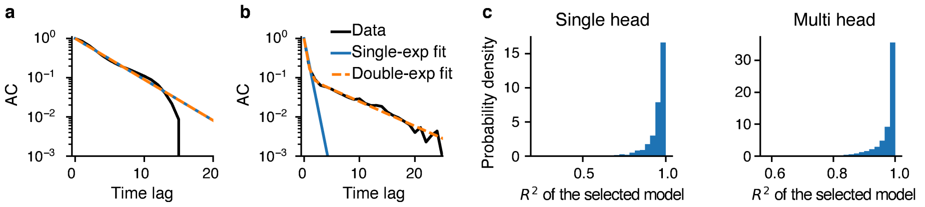


Figure S2: ACs of neurons are well captured with single- or double-exponential fits. Note that the y-axis is in logarithmic coordinates, meaning that deviations between the fit and data AC are much smaller in the AC tail compared to initial time lags. **a, b.** Fitting double and single exponential functions to the AC of example (a) single-timescale ($\tau_{\text{net}} = \tau$) and (b) double timescale ($\tau_{\text{net}} > \tau$) neurons. **c.** Values of coefficient of determination R^2 estimated for all selected fits using AIC are close to 1, indicating a good fit.

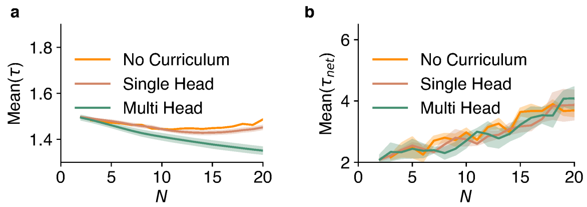


Figure S3: Networks trained without curriculum have similar single-neuron (**a**) and network-mediated (**b**) timescales to networks trained with the single-head curriculum in the range of N that the no-curriculum-trained networks can learn.

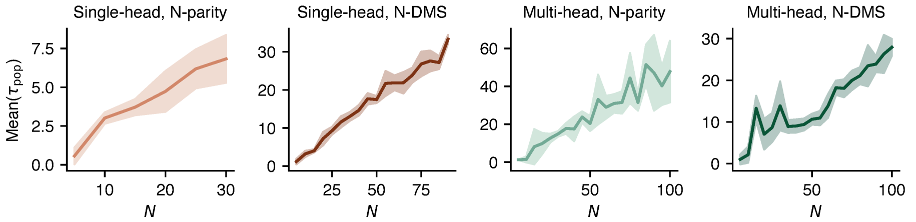


Figure S4: Dependence of population activity timescales τ_{pop} on N. For both tasks and curriculum, the timescale of population activity fluctuations increases with N, indicating a general trend toward slower collective dynamics for tasks with larger memory requirements. Shade - ± STD across 4 networks.

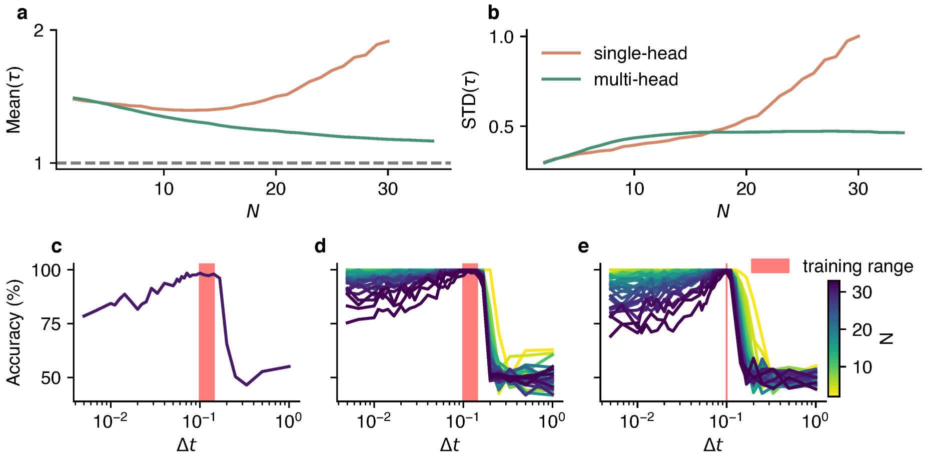


Figure S5: Impact of discretization time-step Δt on training performance. We train networks with $\Delta t = \{\frac{1}{10}, \frac{1}{9}, \frac{1}{8}, \frac{1}{7}\}$, while presenting each input digit for the duration of $T = 1$. **(a)** Similar to discrete-time networks ($\Delta t = 1$), the mean of single-neuron timescales τ increases with N for single-head networks and decreases towards $T = 1$ for multi-head networks. **(b)** The standard deviation of τ indicates heterogeneous τs for single-head networks but constrained values for multi-head networks. **(c,d)** Single-head (c) and multi-head (d) networks can solve the task above the chance level for Δt smaller than their training regime (indicated by the red rectangle) when trained with multiple Δt. **(e)** The networks are slightly more inaccurate when trained on only a single $\Delta t = \frac{1}{10}$. Lines and the color bar indicate different N.

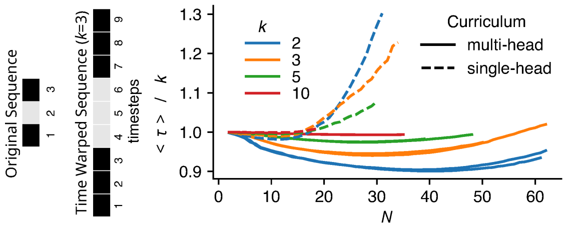


Figure S6: Changing the duration of input presentation. Each input digit is presented to RNN for a duration of $T = k\Delta t$, $\Delta = 1$. Single-neuron timescales (τs) normalized by k remain roughly constant in multi-head networks (i.e. $\tau \to k\Delta t$), but increase with N in single-head networks (cf. Fig. 4b).

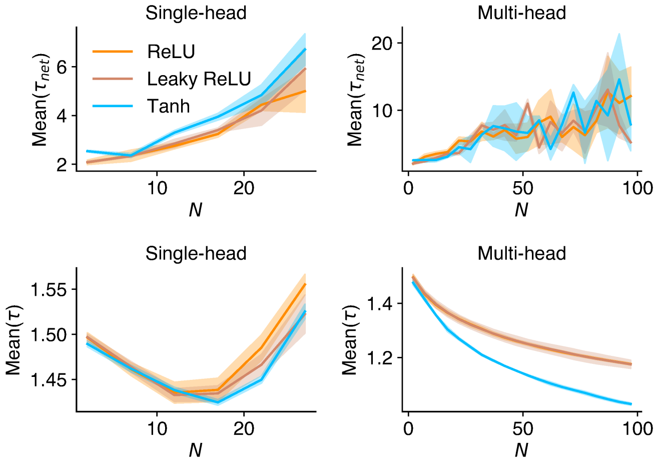


Figure S7: The development of timescales follows similar trajectories when the self-interaction is inside (nonlinear τ) or outside (linear τ) the nonlinearity (leaky-ReLU). Top: network-mediated timescales, bottom: single-neuron timescales. Shades - ± STD.

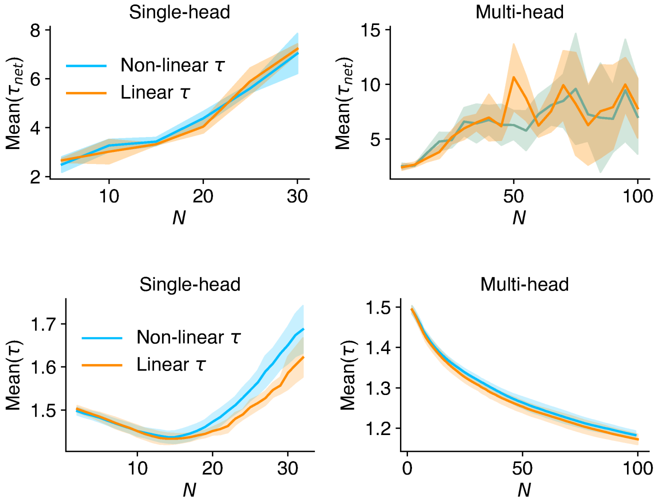


Figure S8: The development of timescales in networks with different nonlinearities follows similar trajectories. Top: network-mediated timescales, bottom: single-neuron timescales. Shades - ± STD.

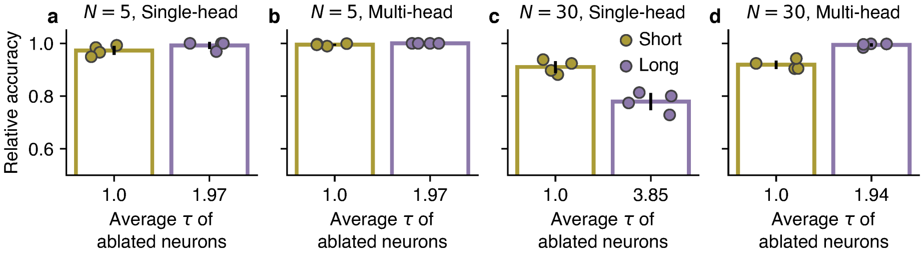


Figure S9: Impact of ablating neurons with distinct timescales on RNNs' performance when neural self-interactions are linear (cf. Fig. 7a,b). **a, b.** Ablating the longest and shortest timescale neurons has minimal effect on network performance when N is small for both curricula. **c, d.** For higher N, ablating long timescale neurons largely decreases the performance of single-head networks, while multi-head networks are more affected by the ablation of short-timescale neurons. Bars - mean, error bars - STD, dots - 4 individual networks.

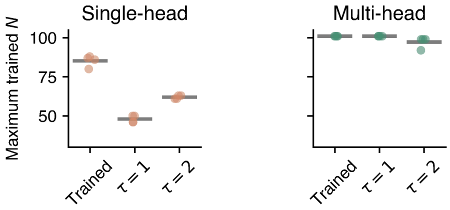


Figure S10: The maximum N solved in the N-DMS task after 1000 epochs (reaching an accuracy of 98%). Similar to the N-parity task (cf. Fig. 4a), models trained with a single-head curriculum rely more on training τ than the multi-head curriculum networks, which prefer to have a small τ and are more agnostic to it being trainable. Horizontal bars - mean.

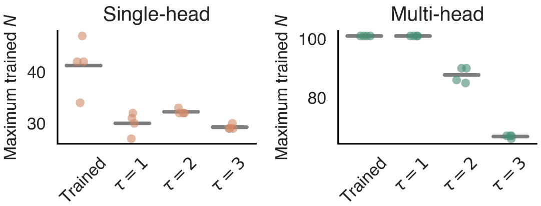


Figure S11: The maximum N solved in the N-parity task (for models with the self-interaction outside the nonlinearity) after 1000 epochs (reaching an accuracy of 98%). In the single-head curriculum, models rely on training τ, whereas in the multi-head curriculum, having τs fixed at 1 value is as good as training them. Results are consistent with the models where the self-interaction is inside the nonlinearity (cf. Fig. 4), Horizontal bars - mean.

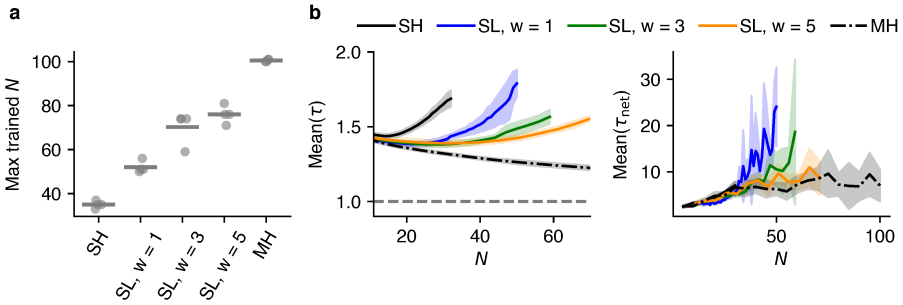


Figure S12: The behavior of networks trained with multi-head-sliding curriculum depends on the size of the sliding window and lies in between extreme curricula. **a.** The maximal trained N (with $> 98\%$ accuracy, within 1000 training epochs) for multi-head-sliding (SL) lies between single-head (SH) and multi-head (MH) networks and increases with the size of sliding window (w). Dots indicate individual networks (4 networks) and the horizontal bars indicate the mean value. **b.** Single-neuron (τ) and network-mediated (τ_{net}) timescales increase with N, but the pace of change reduces as the sliding window grows. Shadings indicate ± std computed across 4 trained networks.

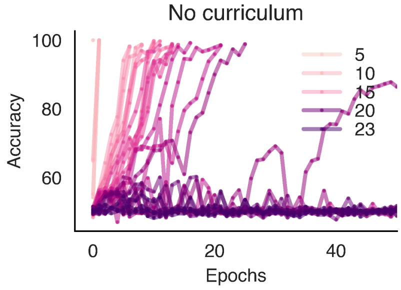


Figure S13: Training without a curriculum on the N-DMS task. For each N, 4 models are independently trained for 50 epochs or until reaching $> 98\%$ accuracy. Similar to the N-Parity task (cf. Fig. 3a), the ability to solve the task decreases as we increase N. From $N > 20$, we see the network is no longer capable of finding a solution to the task even with longer training time.

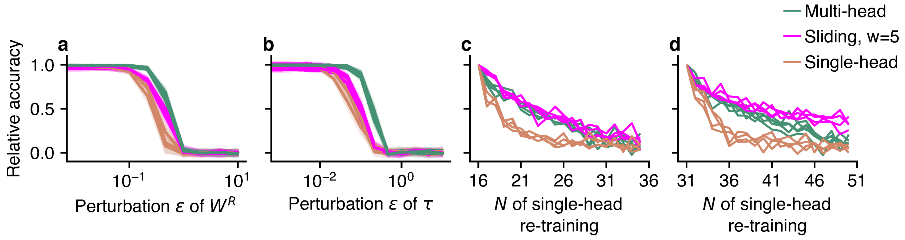


Figure S14: Robustness of networks trained with multi-head-sliding curriculum. **a, b.** Multi-head-sliding networks are more robust than single-head networks but less robust than the multi-head networks against perturbations of recurrent connectivity (a) and trained timescale τ (b). Each line indicates one trained network (4 networks for each curriculum). Shades indicate ± std computed across 10 trials. **c, d.** retraining of networks trained with different curricula as a single-head network on new Ns (for 20 epochs). Multi-head-sliding networks achieve higher relative accuracy when retrained for a higher N in comparison to single-head networks. If originally trained for small Ns (c, $N = 16$), they have similar retraining accuracy to multi-head networks, but for larger Ns (d, $N = 31$) their accuracy suppresses the multi-head networks. Each line indicates the relative accuracy for one network (4 networks for each curriculum).

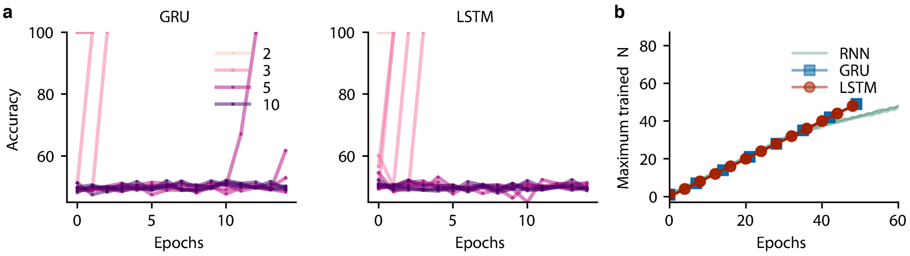


Figure S15: Comparing the impact of curriculum on different recurrent architectures. Two different architectures, GRU and LSTM, are trained on the N-Parity task with and without a curriculum. We observe that the GRU and LSTM both exhibit instability when training without a curriculum (**a**), but are comparable to the RNNs with the multi-head curriculum (**b**).

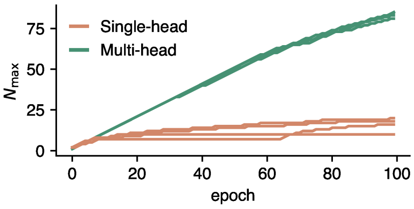


Figure S16: Comparison of single- and multi-head curricula for training LSTMs on N-parity task. Networks trained with the multi-head curriculum can reach a higher N faster than networks trained with the single-head curriculum.

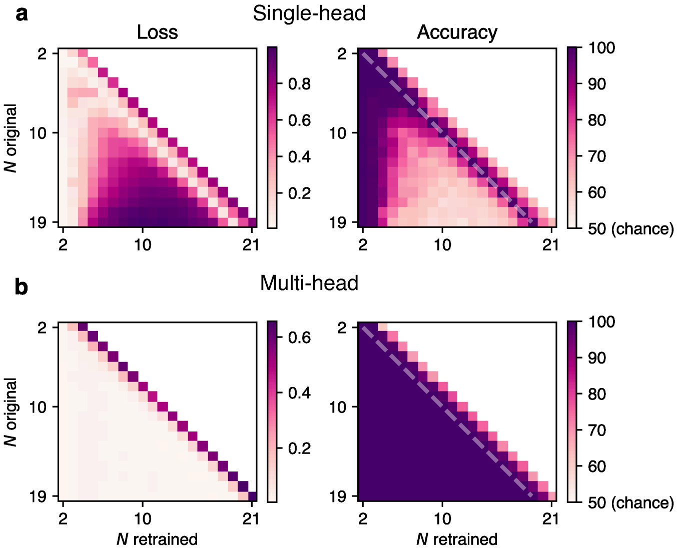


Figure S17: Single-head **(a)** and multi-head **(b)** networks loaded for $N \in [2, \ldots, 19]$ have new readout heads retrained on new tasks with $N^* \in [2, \ldots, N+2]$. The heat map of the loss and accuracy of these retrained networks (after a maximum of 10 epochs or reaching an accuracy of 98%+) shows the robustness of the multi-head networks to catastrophic forgetting, as well as an improvement towards forward compatibility in the $N^* > N$ region. The dotted line indicates the diagonal.

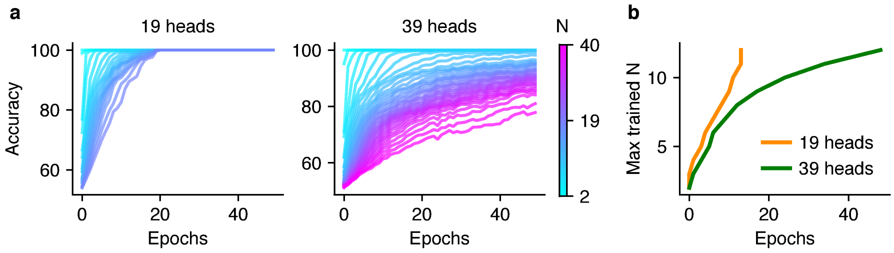


Figure S18: Emergence of curriculum during multi-head training. **a.** In the absence of an explicit curriculum, multi-head networks solve smaller Ns before solving the large Ns. The color bar indicates the range of Ns. **b.** The speed of training reduces with the increasing number of heads. The network with 19 heads needs fewer epochs to solve the same N (i.e. reaching 98% accuracy) than the network with 39 heads.

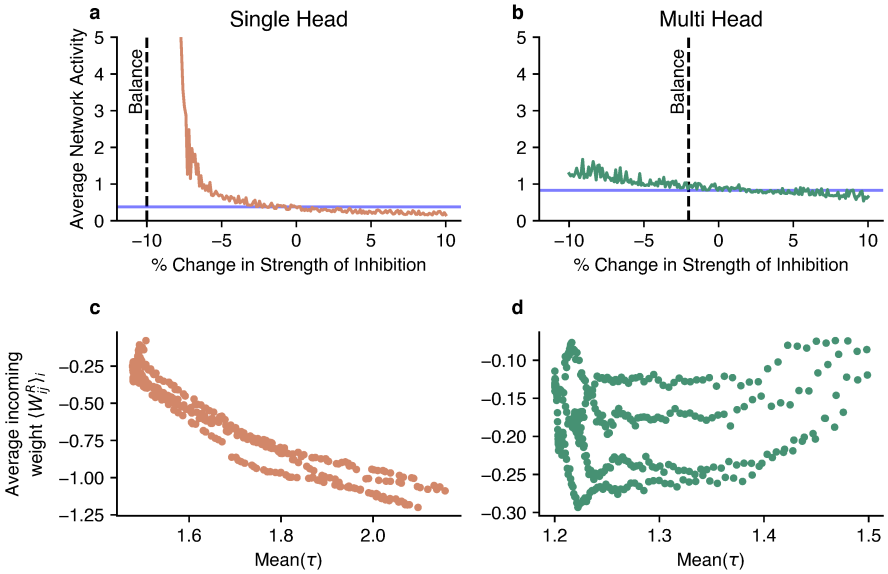


Figure S19: Networks trained with the single-head curriculum require strong inhibitory connectivity. We perturb the inhibitory (negative) connections of a single **(a)** and a multi-head network **(b)**. We see that the activity of the single-head network explodes as we approach the balanced point (the point where the average of incoming weights becomes 0, indicated by the horizontal blue line). On the contrary, the multi-head network is quite robust and produces activity within a normal range even after the balanced point. **(c)** In the single-head networks, we observe a negative correlation between the average τ and the average strength of incoming weights for each neuron (i.e. higher τ is correlated with more negative average weight). This relationship is not present for multi-head networks **(d)** that are largely balanced and maintain small τ.

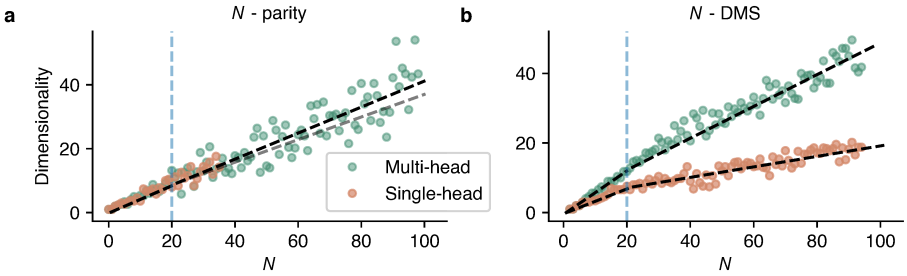


Figure S20: Dimensionality of activity increases approximately linearly with N for the N-parity task and sub-linearly for the N-DMS task. We separately fit the data points for $N \in [0, \ldots, 20]$ and $N \in [20, \ldots, 100]$ ($N \in [20, \ldots, 30]$ for the single-head N-parity network) and we observe that in the N-parity task, the two lines largely coincide, while in the N-DMS case there is a clear change in the slope of the line, suggesting a sub-linear increase with N.

A.5 Environmental variability and network structure determine the optimal plasticity mechanisms in embodied agents

Environmental variability and network structure determine the optimal plasticity mechanisms in embodied agents

Emmanouil Giannakakis[1,2], Sina Khajehabdollahi [1] and Anna Levina[1,2,3]

[1]Department of Computer Science, University of Tübingen, Tübingen, Germany
[2]Max Planck Institute for Biological Cybernetics, Tübingen, Germany
[3]Bernstein Center for Computational Neuroscience Tübingen, Tübingen, Germany

Abstract

The evolutionary balance between innate and learned behaviors is highly intricate, and different organisms have found different solutions to this problem. We hypothesize that the emergence and exact form of learning behaviors is naturally connected with the statistics of environmental fluctuations and tasks an organism needs to solve. Here, we study how different aspects of simulated environments shape an evolved synaptic plasticity rule in static and moving artificial agents. We demonstrate that environmental fluctuation and uncertainty control the reliance of artificial organisms on plasticity. Interestingly, the form of the emerging plasticity rule is additionally determined by the details of the task the artificial organisms are aiming to solve. Moreover, we show that co-evolution between static connectivity and interacting plasticity mechanisms in distinct sub-networks changes the function and form of the emerging plasticity rules in embodied agents performing a foraging task.

Introduction

One of the defining features of living organisms is their ability to adapt to their environment and incorporate new information to modify their behavior. It is unclear how the ability to learn first evolved (Papini, 2012), but its utility appears evident. Natural environments are too complex for all the necessary information to be hardcoded genetically (Snell-Rood, 2013) and more importantly, they keep changing during an organism's lifetime in ways that cannot be anticipated (Ellefsen, 2014; Dunlap and Stephens, 2016). The link between learning and environmental uncertainty and fluctuation has been extensively demonstrated in both natural (Kerr and Feldman, 2003; Snell-Rood and Steck, 2019), and artificial environments (Nolfi and Parisi, 1996).

Nevertheless, the ability to learn does not come without costs. For the capacity to learn to be beneficial in evolutionary terms, a costly nurturing period is often required, a phenomenon observed in both biological (Thornton and Clutton-Brock, 2011), and artificial organisms (Eskridge and Hougen, 2012). Additionally, it has been shown that in some complex environments, hardcoded behaviors may be superior to learned ones given limits in the agent's lifetime and environmental uncertainty (Dunlap and Stephens, 2009; Fawcett et al., 2012; Lange and Sprekeler, 2020).

The theoretical investigation of the optimal balance between learned and innate behaviors in natural and artificial systems goes back several decades. However, it has recently found also a wide range of applications in applied AI systems (Lee and Lee, 2020; Biesialska et al., 2020). Most AI systems are trained for specific tasks, and have no need for modification after their training has been completed. Still, technological advances and the necessity to solve broad families of tasks make discussions about life-like AI systems relevant to a wide range of potential application areas. Thus the idea of open-ended AI agents (Open Ended Learning Team et al., 2021) that can continually interact with and adapt to changing environments has become particularly appealing.

Many different approaches for introducing lifelong learning in artificial agents have been proposed. Some of them draw direct inspiration from actual biological systems (Schmidhuber, 1987; Parisi et al., 2019). Among them, the most biologically plausible solution is to equip artificial neural networks with some local neural plasticity (Thangarasa et al., 2020), similar to the large variety of synaptic plasticity mechanisms (Citri and Malenka, 2008; Feldman, 2009; Caroni et al., 2012) that performs the bulk of the learning in the brains of living organisms (Magee and Grienberger, 2020). The artificial plasticity mechanisms can be optimized to modify the connectivity of the artificial neural networks toward solving a particular task. The optimization can use a variety of approaches, most commonly evolutionary computation.

The idea of meta-learning or optimizing synaptic plasticity rules to perform specific functions has been recently established as an engineering tool that can compete with state-of-the-art machine learning algorithms on various complex tasks (Burms et al., 2015; Najarro and Risi, 2020; Pedersen and Risi, 2021; Yaman et al., 2021). Additionally, it can be used to reverse engineer actual plasticity mechanisms found in biological neural networks and uncover their functions (Confavreux et al., 2020; Jordan et al., 2021).

Here, we study the effect that different factors (environmental fluctuation and reliability, task complexity) have on the form of evolved functional reward-modulated plasticity rules. We investigate the evolution of plasticity rules in static, single-layer simple networks. Then we increase the complexity by switching to moving agents performing a complex foraging task. In both cases, we study the impact of different environmental parameters on the form of the evolved plasticity mechanisms and the interaction of learned and static network connectivity. Interestingly, we find that different environmental conditions and different combinations of static and plastic connectivity have a very large impact on the resulting plasticity rules.

Methods

Environment

We imagine an agent who must forage to survive in an environment presenting various types of complex food particles. Each food particle is composed of various amounts and combinations of N ingredients that can have positive (food) or negative (poison) values. The value of a food particle is a weighted sum of its ingredients. To predict the reward value of a given resource, the agent must learn the values of these ingredients by interacting with the environment. The priors could be generated by genetic memory, but the exact values are subject to change.

To introduce environmental variability, we stochastically change the values of the ingredients. More precisely, we define two ingredient-value distributions E_1 and E_2 (Guttenberg, 2019) and switch between them, with probability p_{tr} for every time step. We control how (dis)similar the environments are by parametrically setting $E_2 = (1-2d_e)E_1$, with $d_e \in [0,1]$ serving as a distance proxy for the environments; when $d_e = 0$, the environment remains unchanged, and when $d_e = 1$ the value of each ingredient fully reverses when the environmental transition happens. For simplicity, we take values of the ingredients in E_1 equally spaced between -1 and 1 (for the visualization, see Fig. 3a, b).

Static agent

The static agent receives passively presented food as a vector of ingredients and can assess its compound value using the linear summation of its sensors with the (learned or evolved) weights, see Fig. 1. The network consists of N sensory neurons that are projecting to a single post-synaptic neuron. At each time step, an input $X_t = (x_1, \dots, x_N)$ is presented, were the value x_i, $i \in \{1, \dots, N\}$ represents the quantity of the ingredient i. We draw x_i independently form a uniform distribution on the $[0,1]$ interval ($x_i \sim U(0,1)$). The value of each ingredient w_i^c is determined by the environment (E_1 or E_2).

The postsynaptic neuron outputs a prediction of the food X_t value as $y_t = g(WX_t^T)$. Throughout the paper, g will be either the identity function, in which case the prediction neuron is linear, or a step-function; however, it could be any other nonlinearity, such as a sigmoid or ReLU. After outputting the prediction, the neuron receives feedback in the form of the real value of the input R_t. The real value is computed as $R_t = W^c X_t^T + \xi$, where $W^c = (w_1^c, \dots, w_N^c)$ is the actual value of the ingredients, and ξ is a term summarizing the noise of reward and sensing system $\xi \sim \mathcal{N}(0, \sigma)$.

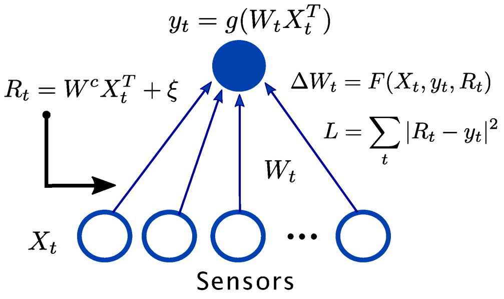


Figure 1: *An outline of the static agent's network. The sensor layer receives inputs representing the quantity of each ingredient of a given food at each time step. The agent computes the prediction of the food's value y_t and is then given the true value R_t; it finally uses this information in the plasticity rule to update the weight matrix.*

For the evolutionary adjustment of the agent's parameters, the loss of the static agent is the sum of the mean squared errors (MSE) between its prediction y_t and the reward R_t over the lifetime of the agent. The agent's initial weights are set to the average of the two ingredient value distributions, which is the optimal initial value for the case of symmetric switching of environments that we consider here.

Moving Agent

As a next step, we incorporate the sensory network of static agents into embodied agents that can move around in an environment scattered with food. To this end, we merge the static agent's network with a second, non-plastic motor network that is responsible for controlling the motion of the agent in the environment. Specifically, the original plastic network now provides the agent with information about the value of the nearest food. The embodied agent has additional sensors for the distance from the nearest food, the angle between the current velocity and the nearest food direction, its own velocity, and its own energy level (sum of consumed food values). These inputs are processed by two hidden layers (of 30 and 15 neurons) with $\tanh$ activation. The network's outputs are angular and linear acceleration, Fig. 2.

The embodied agents spawn in a 2D space with periodic boundary conditions along with a number of food particles that are selected such that the mean of the food value distribution is ~ 0. An agent can eat food by approaching it sufficiently closely, and each time a food particle is eaten,

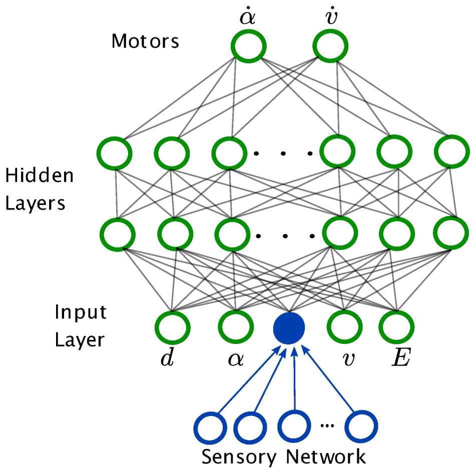


Figure 2: *An outline of the network controlling the foraging agent. The sensor layer receives inputs at each time step (the ingredients of the nearest food), which are processed by the plastic layer in the same way as the static sensory network, Fig. 1. The output of that network is given as input to the motor network, along with the distance d and angle α to the nearest food, the current velocity v, and energy E of the agent. These signals are processed through two hidden layers to the final output of motor commands as the linear and angular acceleration of the agent*

it is re-spawned with the same value somewhere randomly on the grid (following the setup of (Khajehabdollahi et al., 2022)). After 5000 time steps, the cumulative reward of the agent (the sum of the values of all the food it consumed) is taken as its fitness, at which point the best agents are selected by the genetic algorithm and used to initialize the next generation. The environment (food and agents' positions) is re-initialized at the start of each generation. During the evolutionary optimization, the parameters for both the motor network (connections) and plastic network (learning rule parameters) are evolved simultaneously (the genotype includes both motor weights and plasticity parameters), and so agents must learn to move and discriminate good/bad food at the same time.

Plasticity rule parametrization

Reward-modulated plasticity is one of the most promising explanations for biological credit assignment (Legenstein et al., 2008). In our network, the plasticity rule that updates the weights of the linear sensor network is a reward-modulated rule which is parameterized as a linear combination of the input, the output, and the reward at each time step:

$$\Delta W_t = \eta_p [R_t \cdot \overbrace{(\theta_1 X_t y_t + \theta_2 y_t + \theta_3 X_t + \theta_4)}^{\text{Reward Modulated}} + \underbrace{(\theta_5 X_t y_t + \theta_6 y_t + \theta_7 X_t + \theta_8)}_{\text{Hebbian}}]. \quad (1)$$

Additionally, after each plasticity step, the weights are normalized by mean subtraction, an important step for the stabilization of Hebbian-like plasticity rules (Zenke and Gerstner, 2017).

We use a genetic algorithm to optimize the learning rate η_p and amplitudes of different terms $\theta = (\theta_1, \ldots, \theta_8)$. The successful plasticity rule after many food presentations must converge to a weight vector that predicts the correct food values (or allows the agent to correctly decide whether to eat a food or avoid it).

To have comparable results, we divide $\theta = (\theta_1, \ldots, \theta_8)$ by $\theta_{\max} = \max_k |\theta_k|$. So that $\theta/\theta_{\max} = \theta^{\text{norm}} \in [-1, 1]^8$. We then multiply the learning rate η_p with $\theta_{\max}$ to maintain the rule's evolved form unchanged, $\eta_p^{\text{norm}} = \eta_p \cdot \theta_{\max}$. In the following, we always use normalized η_p and θ, omitting $^{\text{norm}}$.

Evolutionary Algorithm

To evolve the plasticity rule and the moving agents' motor networks, we use a simple genetic algorithm with elitism (Deb, 2011). The agents' parameters are initialized at random (drawn from a Gaussian distribution), then the sensory network is trained by the plasticity rule and finally, the agents are evaluated. After each generation, the best-performing agents (top 10 % of the population size) are selected and copied into the next generation. The remaining 90 % of the generation is repopulated with mutated copies of the best-performing agents. We mutate agents by adding independent Gaussian noise ($\sigma = 0.1$) to its parameters. Unless specified otherwise, we train a population of 100 agents for 200 generations.

Results

Environmental and reward variability control the evolved learning rates of the static agents

To start with, we consider a static agent whose goal is to identify the value of presented food correctly. The static reward-prediction network quickly evolves the parameters of the learning rule, successfully solving the prediction task. We first look at the evolved learning rate η_p, which determines how fast (if at all) the network's weight vector is updated during the lifetime of the agents. We identify three factors that control the learning rate parameter the EA converges to: the distance between the environments, the noisiness of the reward, and the rate of environmental transition.

The first natural factor is the distance d_e between the two environments, with a larger distance requiring a higher

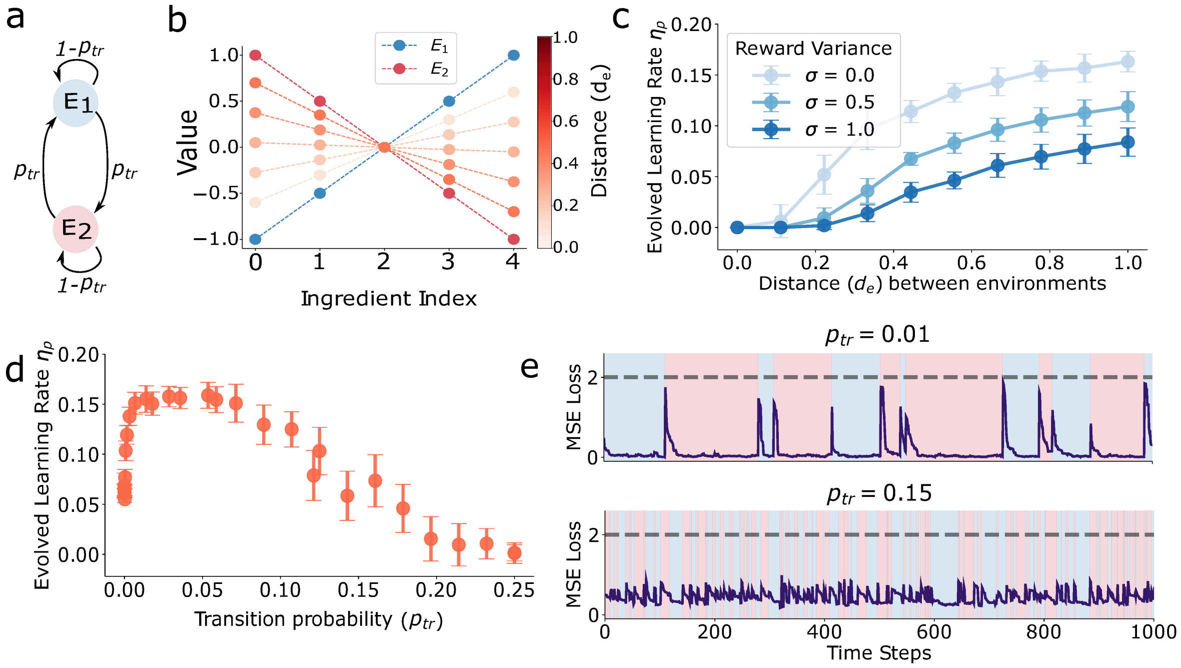


Figure 3: ***a.*** *Schematic representation of two-states Markov model with transition probability p_{tr} between two environments E_1 and E_2 defined by the ingredient value distributions.* ***b.*** *We vary the E_2 environment by changing the ingredient values linearly $E_2 = (1 - 2d_e)E_1$, the d_e is indicated by the color.* ***c.*** *The evolved learning rate η_p grows with the distance d_e between the environments and decreases with the reward variance σ.* ***d.*** *The environment transition probability p_{tr} (here for $d_e = 1$ and $\sigma = 0.25$) has a non-monotonous relationship with the evolved learning rate η_p. Up to a certain point, more rapid transitions lead to faster learning, but too rapid environmental transition leads to a reduction of the evolved learning rate.* ***e.*** *For slow environmental transition (top), the agent fully adapts to the environment after each transition. If the transitions happen fast (bottom), the agent maintains an intermediate position between the two environments and never fully adapts to either of them.*

learning rate, Fig. 3c. This is an expected result since the convergence time to the “correct” weights is highly dependent on the initial conditions. If an agent is born at a point very close to optimality, which naturally happens if the environments are similar, the distance it needs to traverse on the fitness landscape is small. Therefore it can afford to have a small learning rate, which leads to a more stable convergence and is not affected by noise.

A second parameter that impacts the learning rate is the variance of the rewards. The reward an agent receives for the plasticity step contains a noise term ξ that is drawn from a zero mean Gaussian distribution with standard deviation σ. This parameter controls the unreliability of the agent’s sensory system, i.e., higher σ means that the information the agent gets about the value of the foods it consumes cannot be fully trusted to reflect the actual value of the foods. As σ increases, the learning rate η_p decreases, which means that the more unreliable an environment becomes, the less an agent relies on plasticity to update its weights, Fig. 3c. Indeed for some combinations of relatively small distance d_e and high reward variance σ, the EA converges to a learning rate of $\eta_p \approx 0$. This means that the agent opts to have no adaptation during its lifetime and remain at the mean of the two environments. It is an optimal solution when the expected loss due to ignoring the environmental transitions is, on average, lower than the loss the plastic network will incur by learning via the (often misleading because of the high σ) environmental cues.

A final factor that affects the learning rate the EA will converge to is the frequency of environmental change during an agent’s lifetime. Since the environmental change is modeled as a simple, two-state Markov process (Fig. 3a), the control parameter is the transition probability p_{tr}.

When keeping everything else the same, the learning rate rapidly rises as we increase the transition probability from 0, and after reaching a peak, it begins to decline slowly, eventually reaching zero (Fig. 3d). This means that when environmental transition is very rare, agents opt for a very

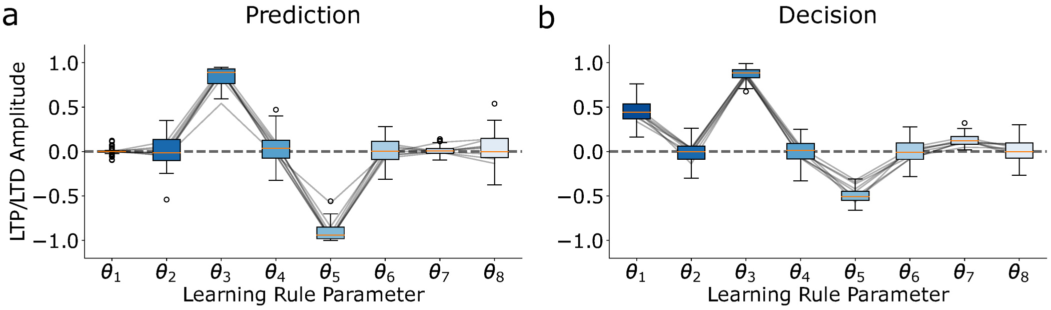


Figure 4: *The evolved parameters $\theta = (\theta_1, \ldots, \theta_8)$ of the plasticity rule for the reward prediction (**a.**) and the decision (**b.**) tasks, for a variety of parameters ($p_{tr} = 0.01$, $d_e \in 0, 0.1, \ldots, 1$, and $\sigma \in 0, 0.1, \ldots, 1$ in all 100 combinations). Despite the relatively small difference between the tasks, the evolved learning rules differ considerably. For visual guidance, the lines connect θs from the same run.*

low learning rate, allowing a slow and stable convergence to an environment-appropriate weight vector that leads to very low losses while the agent remains in that environment. As the rate of environmental transition increases, faster learning is required to speed up convergence in order to exploit the (comparatively shorter) stays in each environment. Finally, as the environmental transition becomes too fast, the agents opt for slower or even no learning, which keeps them near the middle of the two environments, ensuring that the average loss of the two environments is minimal (Fig. 3d).

The form of the evolved learning rule depends on the task: Decision vs. Prediction

The plasticity parameters $\theta = (\theta_1, \ldots, \theta_8)$ for the reward-prediction task converge on approximately the same point, regardless of the environmental parameters (Fig. 4a). In particular, $\theta_3 \to 1$, $\theta_5 \to -1$, $\theta_i \to 0$ for all other i, and thus the learning rule converges to:

$$\Delta W_t = \eta_p[\theta_3 X_t R_t + \theta_5 X_t y_t] \approx \eta_p X_t(R_t - y_t). \tag{2}$$

Since by definition $y_t = g(W_t X_t^T) = W_t X_t^T$ ($g(x) = x$ in this experiment) and $R_t = W^c X_t^T + \xi$ we get:

$$\Delta W_t = \eta_p X_t(W^c - W_t)X_t^T + \eta_p \xi X_t^T. \tag{3}$$

Thus the distribution of ΔW_t converges to a distribution with mean 0 and variance depending on η_p and σ and W converges to W^c. So this learning rule will match the agent's weight vector with the vector of ingredient values in the environment.

We examine the robustness of the learning rule the EA discovers by considering a slight modification of our task. Instead of predicting the expected food value, the agent now needs to decide whether to eat the presented food or not. This is done by introducing a step-function nonlinearity ($g(x) = 1$ if $x \geq 1$ and 0 otherwise). Then the output $y(t)$ is computed as:

$$y_t = \begin{cases} 1, & \text{if } W_t X_t^T \geq 0, \\ 0, & \text{if } W_t X_t^T < 0. \end{cases} \tag{4}$$

Instead of the MSE loss between prediction and actual value, the fitness of the agent is now defined as the sum of the food values it chose to consume (by giving $y_t = 1$). Besides these two changes, the setup of the experiments remains exactly the same.

The qualitative relation between η_p and parameters of environment d_e, σ and p_{tr} is preserved in the changed experiment. However, the resulting learning rule is significantly different (Fig. 4). The evolution converges to the following learning rule:

$$\Delta W_t = \begin{cases} \eta_p X_t[\theta_3 R_t + \theta_7], \ y_t = 0, \\ \eta_p X_t[(\theta_1 + \theta_3)R_t + (\theta_5 + \theta_7)], \ y_t = 1. \end{cases} \tag{5}$$

In both cases, the rule has the form $\Delta W_t = \eta_p X_t[\alpha_y R_t + \beta_y]$. Thus, the ΔW_t is positive or negative depending on whether the reward R_t is above or below a threshold ($\gamma = -\beta_y/\alpha_y$) that depends on the output decision of the network ($y_t = 0$ or 1).

Both learning rules have a clear Hebbian form (coordination of pre- and post-synaptic activity) and use the incoming reward signal as a threshold. These similarities indicate some common organizing principles of reward-modulated learning rules, but their significant differences highlight the sensitivity of the optimization process to task details.

The learning rate of embodied agents depends on environmental variability

We now turn to the moving embodied agents in the 2D environment. To optimize these agents, both the motor network's connections and the sensory network's plasticity parameters

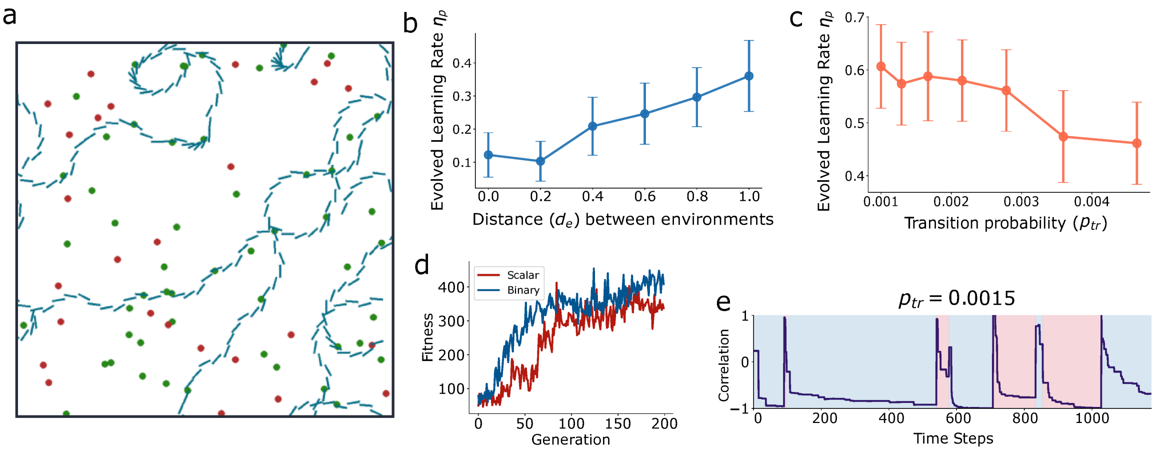


Figure 5: ***a.*** *The trajectory of an agent (blue line) in the 2D environment. A well-trained agent will approach and consume food with positive values (green dots) and avoid negative food (red dots).* ***b.*** *The learning rate of the plastic sensory network eta_p grows with the distance between environments d_e* ***c.*** *and decreases with the frequency of environmental change.* ***d.*** *The fitness of an agent (measured as the total food consumed over its lifetime) increases over generations of the EA for both the scalar and binary readouts in the sensory network.* ***e.*** *The Pearson correlation coefficient of an evolved agent's weights with the ingredient value vector of the current environment (E_1 - blue, E_2 - red). In this example, the agent's weights are anti-correlated with its environment, which is not an issue for performance since the motor network can interpret the inverted signs of food.*

evolve simultaneously. Since the motor network is initially random and the agent has to move to find food, the number of interactions an agent experiences in its lifetime can be small, slowing down the learning. However, having the larger motor network also has benefits for evolution because it allows the output of the plastic network to be read out and transformed in different ways, resulting in a broad set of solutions.

The agents can solve the task effectively by evolving a functional motor network and a plasticity rule that converges to interpretable weights (Fig. 5a). After $\sim$ 100 evolutionary steps (Fig. 5d), the agents can learn the ingredient value distribution using the plastic network and reliably move towards foods with positive values while avoiding the ones with negative values.

We compare the dependence of the moving and the static agents on the parameters of the environment: d_e and the state transition probability p_{tr}. At first, in order to simplify the experiment, we set the transition probability to 0, but fixed the initial weights to be the average of E_1 and E_2, while the real state is E_2. In this experiment, the distance between states d_e indicates twice the distance between the agent's initial weights and the optimal weights (the environment's ingredient values) since the agent is initialized at the mean of the two environment distributions. Same as for the static agent, the learning rate increases with the distance d_e (Fig. 5b).

Then, we examine the effect of the environmental transition probability p_{tr} on the evolved learning rate η_p. In order for an agent to get sufficient exposure to each environment, we scale down the probability p_{tr} from the equivalent experiment for the static agents. We find that as the probability of transition increases, the evolved learning rate η_p decreases (Fig. 5c). This fits with the larger trend for the static agent, although there is a clear difference when it comes to the increase for very small transition probabilities that were clearly identifiable in the static but not the moving agents. This could be due to much sparser data and possibly the insufficiently long lifetime of the moving agent (the necessity of scaling makes direct comparisons difficult). Nevertheless, overall we see that the associations observed in the static agents between environmental distance d_e and transition probability p_{tr} and the evolved learning rate η_p are largely maintained in the moving agents. Still, more data would be needed to make any conclusive assertions about the exact effect of these environmental parameters on the emerging plasticity mechanisms.

Rule redundancy in the embodied agents

A crucial difference between the static and the moving agents is the function the plasticity has to perform. While in the static agents, the plasticity has to effectively identify the exact value distribution of the environment in order to produce accurate predictions, in the embodied agents, the plasticity has to merely produce a representation of the environment that the motor network can evolve to interpret

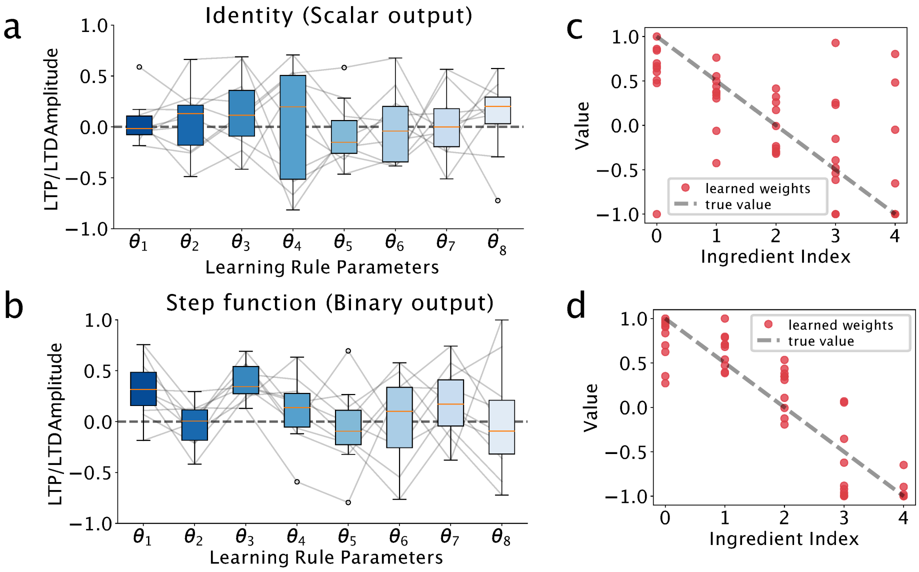


Figure 6: *The evolved parameters of moving agents' plasticity rule for the $g(s) = x$, identity (**a.**) and the step function (Eq. 4) (**b.**) sensory networks (the environmental parameters here are $d_e \in [0,1]$, $\sigma = 0$ and $p_{tr} = 0.001$). The step function (binary output) network evolved a more structured plasticity rule (e.g., $\theta_3 > 0$ for all realizations) than the linear network. Moreover, the learned weights for the identity network (**c.**) have higher variance and correlate significantly less with the environment's ingredient distribution compared to the learned weights for the thresholded network (**d.**)*

adequately enough to make decisions about which food to consume.

To illustrate the difference, we plot the Pearson correlation coefficient between an agent's weights and the ingredient values of the environment it is moving in (Fig. 5e). We use the correlation instead of the MSE loss (which we used for the static agents in Fig. 3e) because the amplitude of the weight vector varies a lot for different agents and meaningful conclusions cannot be drawn from the MSE loss. For many agents, the learned weights are consistently anti-correlated with the actual ingredient values (an example of such an agent is shown in Fig. 5e). This means that the output of the sensory network will have the opposite sign from the actual food value. While in the static network, this would lead to very bad predictions and high loss, in the foraging task, these agents perform exactly as well as the ones where the weights and ingredients values are positively correlated, since the motor network can simply learn to move towards food for which it gets a negative instead of a positive sensory input.

This additional step of the output of the plastic network going through the motor network before producing any behavior has a strong effect on the plasticity rules that the embodied agents evolve. Specifically, if we look at the emerging rules the top performing agents have evolved (Fig. 6a), it becomes clear that, unlike the very well-structured rules of the static agents (Fig. 4a), there is now virtually no discernible pattern or structure. The difference becomes even clearer if we look at the learned weights (at the end of a simulation) of the best-performing agents (Fig. 6c). While there is some correlation with the environment's ingredient value distribution, the variance is very large, and they do not seem to converge on the "correct" values in any way. This is to some extent expected since, unlike the static agents where the network's output has to be exactly correct, driving the evolution of rules that converge to the precise environmental distribution, in the embodied networks, the bulk of the processing is done by the motor network which can evolve to interpret the scalar value of the sensory network's output in a variety of ways. Thus, as long as the sensory network's plasticity rule co-evolves with the motor network, any plasticity rule that learns to produce consistent information about the value of encountered food can potentially be selected.

To further test this assumption, we introduce a bottleneck of information propagation between the sensory and motor networks by using a step-function nonlinearity on the output of the sensory network (Eq. 4). Similarly to the decision task of the static network, the output of the sensory network now becomes binary. This effectively reduces the flow of information from the sensory to the motor network, forcing the sensory network to consistently decide whether food should be consumed (with the caveat that the motor network can still interpret the binary sign in either of two ways, either consuming food marked with 1 or the ones marked with 0 by the sensory network). The agents perform equally well in this variation of the task as before (Fig. 5d), but now, the evolved plasticity rules seem to be more structured (Fig. 6b). Moreover, the variance of the learned weights in the best-performing agents is significantly reduced (Fig. 6d), which indicates that the bottleneck in the sensory network is increasing selection pressure for rules that learn the environment's food distribution accurately.

Discussion

We find that different sources of variability have a strong impact on the extent to which evolving agents will develop neuronal plasticity mechanisms for adapting to their environment. A diverse environment, a reliable sensory system, and a rate of environmental change that is neither too large nor too small are necessary conditions for an agent to be able to effectively adapt via synaptic plasticity. Additionally, we find that minor variations of the task an agent has to solve or the parametrization of the network can give rise to significantly different plasticity rules.

Our results partially extend to embodied artificial agents performing a foraging task. We show that environmental variability also pushes the development of plasticity in such agents. Still, in contrast to the static agents, we find that the interaction of a static motor network with a plastic sensory network gives rise to a much greater variety of well-functioning learning rules. We propose a potential cause of this degeneracy; as the relatively complex motor network is allowed to read out and process the outputs from the plastic network, any consistent information coming out of these outputs can be potentially interpreted in a behaviorally useful way. Reducing the information the motor network can extract from the sensory system significantly limits learning rule variability.

Our findings on the effect of environmental variability concur with the findings of previous studies (Lange and Sprekeler, 2020) that have identified the constraints that environmental variability places on the evolutionary viability of learning behaviors. We extend these findings in a mechanistic model which uses a biologically plausible learning mechanism (synaptic plasticity). We show how a simple evolutionary algorithm can optimize the different parameters of a simple reward-modulated plasticity rule for solving simple prediction and decision tasks. Reward-modulated plasticity has been extensively studied as a plausible mechanism for credit assignment in the brain (Florian, 2007; Baras and Meir, 2007; Legenstein et al., 2008) and has found several applications in artificial intelligence and robotics tasks (Burms et al., 2015; Bing et al., 2019). Here, we demonstrate how such rules can be very well-tuned to take into account different environmental parameters and produce optimal behavior in simple systems.

Additionally, we demonstrate how the co-evolution of plasticity and static functional connectivity in different subnetworks fundamentally changes the evolutionary pressures on the resulting plasticity rules, allowing for greater diversity in the form of the learning rule and the resulting learned connectivity. Several studies have demonstrated how, in biological networks, synaptic plasticity heavily interacts with (Butz et al., 2014; Stampanoni Bassi et al., 2019; Bernáez Timón et al., 2022) and is driven by network topology (Giannakakis et al., 2023). Moreover, it has been recently demonstrated that biological plasticity mechanisms are highly redundant in the sense that any observed neural connectivity or recorded activity can be achieved with a variety of distinct, unrelated learning rules (Ramesh, 2023). This observed redundancy of learning rules in biological settings complements our results and suggests that the function of plasticity rules cannot be studied independently of the connectivity and topology of the networks they are acting on.

The optimization of functional plasticity in neural networks is a promising research direction both as a means to understand biological learning processes and as a tool for building more autonomous artificial systems. Our results suggest that reward-modulated plasticity is highly adaptable to different environments and can be incorporated into larger systems that solve complex tasks.

Future work

This work studies a simplified toy model of neural network learning in stochastic environments. Future work could be built on this basic framework to examine more complex reward distributions and sources of environmental variability. Moreover, a greater degree of biological realism could be added by studying more plausible network architectures (possibly derived from connectomics data) and more sophisticated plasticity rule parametrizations.

Additionally, our foraging simulations were constrained by limited computational resources and were far from exhaustive. Further experiments can investigate environments with different constraints, food distributions, and multiple seasons as well as the inclusion of plasticity on the motor parts of the artificial organisms.

Acknowledgements

This work was supported by a Sofja Kovalevskaja Award from the Alexander von Humboldt Foundation. EG and SK thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for their support. We acknowledge the support from the BMBF through the Tübingen AI Center (FKZ: 01IS18039A). AL is a member of the Machine Learning Cluster of Excellence, EXC number 2064/1 – Project number 39072764.

References

Baras, D. and Meir, R. (2007). Reinforcement Learning, Spike-Time-Dependent Plasticity, and the BCM Rule. *Neural Computation*, 19(8):2245–2279.

Bernáez Timón, L., Ekelmans, P., Konrad, S., Nold, A., and Tchumatchenko, T. (2022). Synaptic plasticity controls the emergence of population-wide invariant representations in balanced network models. *Phys. Rev. Res.*, 4:013162.

Biesialska, M., Biesialska, K., and Costa-jussà , M. R. (2020). Continual lifelong learning in natural language processing: A survey. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics.

Bing, Z., Baumann, I., Jiang, Z., Huang, K., Cai, C., and Knoll, A. (2019). Supervised learning in snn via reward-modulated spike-timing-dependent plasticity for a target reaching vehicle. *Frontiers in Neurorobotics*, 13.

Burms, J., Caluwaerts, K., and Dambre, J. (2015). Reward-modulated hebbian plasticity as leverage for partially embodied control in compliant robotics. *Frontiers in Neurorobotics*, 9.

Butz, M., Steenbuck, I., and van Ooyen, A. (2014). Homeostatic structural plasticity increases the efficiency of small-world networks. *Frontiers in Synaptic Neuroscience*, 6.

Caroni, P., Donato, F., and Muller, D. (2012). Structural plasticity upon learning: regulation and functions. *Nature reviews. Neuroscience*, 13(7):478—490.

Citri, A. and Malenka, R. (2008). Synaptic plasticity: Multiple forms, functions, and mechanisms. *Neuropsychopharmacology : official publication of the American College of Neuropsychopharmacology*, 33:18–41.

Confavreux, B., Zenke, F., Agnes, E., Lillicrap, T., and Vogels, T. (2020). A meta-learning approach to (re)discover plasticity rules that carve a desired function into a neural network. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 16398–16408. Curran Associates, Inc.

Deb, K. (2011). *Multi-objective Optimisation Using Evolutionary Algorithms: An Introduction*, pages 3–34. Springer London, London.

Dunlap, A. S. and Stephens, D. W. (2009). Components of change in the evolution of learning and unlearned preference. *Proceedings of the Royal Society B: Biological Sciences*, 276(1670):3201–3208.

Dunlap, A. S. and Stephens, D. W. (2016). Reliability, uncertainty, and costs in the evolution of animal learning. *Current Opinion in Behavioral Sciences*, 12:73–79. Behavioral ecology.

Ellefsen, K. O. (2014). The evolution of learning under environmental variability. pages 649–656.

Eskridge, B. E. and Hougen, D. F. (2012). Nurturing promotes the evolution of learning in uncertain environments. In *2012 IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, pages 1–6.

Fawcett, T. W., Hamblin, S., and Giraldeau, L.-A. (2012). Exposing the behavioral gambit: the evolution of learning and decision rules. *Behavioral Ecology*, 24(1):2–11.

Feldman, D. E. (2009). Synaptic mechanisms for plasticity in neocortex. *Annual Review of Neuroscience*, 32(1):33–55. PMID: 19400721.

Florian, R. V. (2007). Reinforcement Learning Through Modulation of Spike-Timing-Dependent Synaptic Plasticity. *Neural Computation*, 19(6):1468–1502.

Giannakakis, E., Vinogradov, O., Buendia, V., and Levina, A. (2023). Recurrent connectivity structure controls the emergence of co-tuned excitation and inhibition. *bioRxiv*.

Guttenberg, N. (2019). Evolutionary rates of information gain and decay in fluctuating environments. ALIFE 2019: The 2019 Conference on Artificial Life:365–371.

Jordan, J., Schmidt, M., Senn, W., and Petrovici, M. A. (2021). Evolving interpretable plasticity for spiking networks. *eLife*, 10:e66273.

Kerr, B. and Feldman, M. (2003). Carving the cognitive niche: Optimal learning strategies in homogeneous and heterogeneous environments. *Journal of Theoretical Biology*, 220(2):169–188.

Khajehabdollahi, S., Prosi, J., Giannakakis, E., Martius, G., and Levina, A. (2022). When to Be Critical? Performance and Evolvability in Different Regimes of Neural Ising Agents. *Artificial Life*, 28(4):458–478.

Lange, R. T. and Sprekeler, H. (2020). Learning not to learn: Nature versus nurture in silico.

Lee, C. and Lee, A. (2020). Clinical applications of continual learning machine learning. *The Lancet Digital Health*, 2:e279–e281.

Legenstein, R., Pecevski, D., and Maass, W. (2008). A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback. *PLOS Computational Biology*, 4(10):1–27.

Magee, J. C. and Grienberger, C. (2020). Synaptic plasticity forms and functions. *Annual Review of Neuroscience*, 43(1):95–117. PMID: 32075520.

Najarro, E. and Risi, S. (2020). Meta-learning through hebbian plasticity in random networks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20719–20731. Curran Associates, Inc.

Nolfi, S. and Parisi, D. (1996). Learning to adapt to changing environments in evolving neural networks. *Adaptive Behavior*, 5(1):75–98.

Open Ended Learning Team, Stooke, A., Mahajan, A., Barros, C., Deck, C., Bauer, J., Sygnowski, J., Trebacz, M., Jaderberg, M., Mathieu, M., McAleese, N., Bradley-Schmieg, N., Wong, N., Porcel, N., Raileanu, R., Hughes-Fitt, S., Dalibard, V., and Czarnecki, W. M. (2021). Open-ended learning leads to generally capable agents.

Papini, M. R. (2012). *Evolution of Learning*, pages 1188–1192. Springer US, Boston, MA.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. (2019). Continual lifelong learning with neural networks: A review.

Pedersen, J. W. and Risi, S. (2021). Evolving and merging hebbian learning rules: Increasing generalization by decreasing the number of rules. In *Proceedings of the Genetic and Evolutionary Computation Conference*, GECCO '21, page 892–900, New York, NY, USA. Association for Computing Machinery.

Ramesh, P. (2023). *GANs schön kompliziert: Applications of Generative Adversarial Networks*. PhD thesis, University of Tübingen.

Schmidhuber, J. (1987). Evolutionary principles in self-referential learning. on learning now to learn: The meta-meta-meta...-hook. Diploma thesis, Technische Universitat Munchen, Germany.

Snell-Rood, E. C. (2013). An overview of the evolutionary causes and consequences of behavioural plasticity. *Animal Behaviour*, 85(5):1004–1011. Special Issue: Behavioural Plasticity and Evolution.

Snell-Rood, E. C. and Steck, M. K. (2019). Behaviour shapes environmental variation and selection on learning and plasticity: review of mechanisms and implications. *Animal Behaviour*, 147:147–156.

Stampanoni Bassi, M., Iezzi, E., Gilio, L., Centonze, D., and Buttari, F. (2019). Synaptic plasticity shapes brain connectivity: Implications for network topology. *International Journal of Molecular Sciences*, 20(24).

Thangarasa, V., Miconi, T., and Taylor, G. W. (2020). Enabling continual learning with differentiable hebbian plasticity.

Thornton, A. and Clutton-Brock, T. (2011). Social learning and the development of individual and group behaviour in mammal societies. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 366:978–87.

Yaman, A., Iacca, G., Mocanu, D. C., Coler, M., Fletcher, G., and Pechenizkiy, M. (2021). Evolving Plasticity for Autonomous Learning under Changing Environmental Conditions. *Evolutionary Computation*, 29(3):391–414.

Zenke, F. and Gerstner, W. (2017). Hebbian plasticity requires compensatory processes on multiple timescales. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 372(1715):20160259.

Bibliography

Here are the references in citation order.

[1] Timothy O'Connor. 'Emergent Properties'. In: *The Stanford Encyclopedia of Philosophy*. Ed. by Edward N. Zalta. Winter 2021. Metaphysics Research Lab, Stanford University, 2021 (cited on page 1).

[2] Eric Bonabeau, Marco Dorigo, and Guy Theraulaz. *Swarm intelligence: from natural to artificial systems*. Oxford university press, 1999 (cited on page 1).

[3] Iain D Couzin. 'Collective cognition in animal groups'. In: *Trends in cognitive sciences* 13.1 (2009), pp. 36–43 (cited on page 1).

[4] Miguel A Munoz. 'Colloquium: Criticality and dynamical scaling in living systems'. In: *Reviews of Modern Physics* 90.3 (2018), p. 031001 (cited on page 1).

[5] Luis Gómez-Nava et al. 'Fish shoals resemble a stochastic excitable system driven by environmental perturbations'. In: *Nature Physics* 19.5 (2023), pp. 663–669 (cited on page 1).

[6] Philip W Anderson. 'More Is Different: Broken symmetry and the nature of the hierarchical structure of science.' In: *Science* 177.4047 (1972), pp. 393–396 (cited on page 1).

[7] Per Bak, Chao Tang, and Kurt Wiesenfeld. 'Self-organized criticality: An explanation of the 1/f noise'. In: *Physical review letters* 59.4 (1987), p. 381 (cited on pages 2, 8).

[8] Beno Gutenberg. 'The energy of earthquakes'. In: *Quarterly Journal of the Geological Society* 112.1-4 (1956), pp. 1–14 (cited on pages 2, 8).

[9] William H Press. 'Flicker noises in astronomy and elsewhere'. In: *Comments on Modern Physics, Part C-Comments on Astrophysics, vol. 7, no. 4, 1978, p. 103-119.* 7 (1978), pp. 103–119 (cited on pages 2, 8).

[10] Hans Jacob S Feder and Jens Feder. 'Self-organized criticality in a stick-slip process'. In: *Physical Review Letters* 66.20 (1991), p. 2669 (cited on pages 2, 8).

[11] Vidar Frette et al. 'Avalanche dynamics in a pile of rice'. In: *Nature* 379.6560 (1996), pp. 49–52 (cited on pages 2, 8).

[12] Anna Levina, Udo Ernst, and J Michael Herrmann. 'Criticality of avalanche dynamics in adaptive recurrent networks'. In: *Neurocomputing* 70.10-12 (2007), pp. 1877–1881 (cited on pages 2, 8, 9).

[13] Elad Schneidman et al. 'Weak pairwise correlations imply strongly correlated network states in a neural population'. In: *Nature* 440.7087 (2006), pp. 1007–1012 (cited on page 2).

[14] Fang-Chin Yeh et al. 'Maximum entropy approaches to living neural networks'. In: *Entropy* 12.1 (2010), pp. 89–106 (cited on page 2).

[15] Jonathon Shlens et al. 'The structure of multi-neuron firing patterns in primate retina'. In: *Journal of Neuroscience* 26.32 (2006), pp. 8254–8266 (cited on page 2).

[16] Yasser Roudi, Sheila Nirenberg, and Peter E Latham. 'Pairwise maximum entropy models for studying large biological systems: when they can work and when they can't'. In: *PLoS computational biology* 5.5 (2009), e1000380 (cited on page 2).

[17] Gašper Tkačik et al. 'Thermodynamics and signatures of criticality in a network of neurons'. In: *Proceedings of the National Academy of Sciences* 112.37 (2015), pp. 11508–11513 (cited on pages 2, 25).

[18] John M Beggs. 'The criticality hypothesis: how local cortical networks might optimize information processing'. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 366.1864 (2008), pp. 329–343 (cited on pages 7, 31).

[19] Norman H Packard. 'Adaptation toward the edge of chaos'. In: *Dynamic patterns in complex systems* 212 (1988), pp. 293–301 (cited on page 7).

[20] Chris G Langton. 'Computation at the edge of chaos: Phase transitions and emergent computation'. In: *Physica D: nonlinear phenomena* 42.1-3 (1990), pp. 12–37 (cited on page 7).

[21] Stuart A Kauffman and Sonke Johnsen. 'Coevolution to the edge of chaos: Coupled fitness landscapes, poised states, and coevolutionary avalanches'. In: *Journal of theoretical biology* 149.4 (1991), pp. 467–505 (cited on page 7).

[22] Melanie Mitchell, Peter Hraber, and James P Crutchfield. 'Revisiting the edge of chaos: Evolving cellular automata to perform computations'. In: *arXiv preprint adap-org/9303003* (1993) (cited on page 7).

[23] Joschka Boedecker et al. 'Information processing in echo state networks at the edge of chaos'. In: *Theory in Biosciences* 131 (2012), pp. 205–213 (cited on page 7).

[24] John M Beggs and Dietmar Plenz. 'Neuronal avalanches in neocortical circuits'. In: *Journal of neuroscience* 23.35 (2003), pp. 11167–11177 (cited on pages 7, 31).

[25] Osame Kinouchi and Mauro Copelli. 'Optimal dynamical range of excitable networks at criticality'. In: *Nature physics* 2.5 (2006), pp. 348–351 (cited on page 7).

[26] Woodrow L Shew et al. 'Neuronal avalanches imply maximum dynamic range in cortical networks at criticality'. In: *Journal of neuroscience* 29.49 (2009), pp. 15595–15600 (cited on page 7).

[27] John M Beggs and Nicholas Timme. 'Being critical of criticality in the brain'. In: *Frontiers in physiology* 3 (2012), p. 163 (cited on page 7).

[28] Enzo Tagliazucchi et al. 'Criticality in large-scale brain fMRI dynamics unveiled by a novel point process analysis'. In: *Frontiers in physiology* 3 (2012), p. 20422 (cited on page 7).

[29] James P Sethna. *Statistical mechanics: entropy, order parameters, and complexity*. Vol. 14. Oxford University Press, USA, 2021 (cited on page 7).

[30] Jordan O'Byrne and Karim Jerbi. 'How critical is brain criticality?' In: *Trends in Neurosciences* 45.11 (2022), pp. 820–837 (cited on page 7).

[31] H Stanley. 'Introduction to Phase Transitions and Critical Phenomena Oxford University Press, 1971'. In: *ISBN: 0-19.505316-8 (pbk.)* () (cited on page 7).

[32] Kenneth G Wilson. 'The renormalization group: Critical phenomena and the Kondo problem'. In: *Reviews of modern physics* 47.4 (1975), p. 773 (cited on page 7).

[33] Leo P Kadanoff. 'Scaling and universality in statistical physics'. In: *Physica A: Statistical Mechanics and its Applications* 163.1 (1990), pp. 1–14 (cited on page 7).

[34] H Eugene Stanley. 'Scaling, universality, and renormalization: Three pillars of modern critical phenomena'. In: *Reviews of modern physics* 71.2 (1999), S358 (cited on page 7).

[35] Richard M Levenson et al. 'Pigeons (Columba livia) as trainable observers of pathology and radiology breast cancer images'. In: *PloS one* 10.11 (2015), e0141357 (cited on page 8).

[36] Anthony G Hudetz, Colin J Humphries, and Jeffrey R Binder. 'Spin-glass model predicts metastable brain states that diminish in anesthesia'. In: *Frontiers in systems neuroscience* 8 (2014), p. 234 (cited on page 8).

[37] Enzo Tagliazucchi. 'The signatures of conscious access and its phenomenology are consistent with large-scale brain communication at criticality'. In: *Consciousness and cognition* 55 (2017), pp. 136–147 (cited on page 8).

[38] Gustavo Deco et al. 'Perturbation of whole-brain dynamics in silico reveals mechanistic differences between brain states'. In: *Neuroimage* 169 (2018), pp. 46–56 (cited on page 8).

[39] Heonsoo Lee et al. 'Relationship of critical dynamics, functional connectivity, and states of consciousness in large-scale human brain networks'. In: *Neuroimage* 188 (2019), pp. 228–238 (cited on page 8).

[40] Hyoungkyu Kim and UnCheol Lee. 'Criticality as a determinant of integrated information Φ in human brain networks'. In: *Entropy* 21.10 (2019), p. 981 (cited on page 8).

[41] Tomer Fekete et al. 'Multiscale criticality measures as general-purpose gauges of proper brain function'. In: *Scientific Reports* 11.1 (2021), p. 14441 (cited on page 8).

[42] Daniel Toker et al. 'Consciousness is supported by near-critical slow cortical electrodynamics'. In: *Proceedings of the National Academy of Sciences* 119.7 (2022), e2024455119 (cited on page 8).

[43] Nike Walter and Thilo Hinterberger. 'Determining states of consciousness in the electroencephalogram based on spectral, complexity, and criticality features'. In: *Neuroscience of Consciousness* 2022.1 (2022), niac008 (cited on page 8).

[44] Ariel Haimovici, Pablo Balenzuela, and Enzo Tagliazucchi. 'Dynamical signatures of structural connectivity damage to a model of the brain posed at criticality'. In: *Brain connectivity* 6.10 (2016), pp. 759–771 (cited on page 8).

[45] Luca Cocchi et al. 'Criticality in the brain: A synthesis of neurobiology, models and cognition'. In: *Progress in neurobiology* 158 (2017), pp. 132–152 (cited on page 8).

[46] Vincent Zimmern. 'Why brain criticality is clinically relevant: a scoping review'. In: *Frontiers in neural circuits* 14 (2020), p. 565335 (cited on page 8).

[47] Rodrigo P Rocha et al. 'Recovery of neural dynamics criticality in personalized whole-brain models of stroke'. In: *Nature Communications* 13.1 (2022), p. 3683 (cited on page 8).

[48] Jorge Hidalgo et al. 'Information-based fitness and the emergence of criticality in living systems'. In: *Proceedings of the National Academy of Sciences* 111.28 (2014), pp. 10095–10100 (cited on page 8).

[49] Andrea Roli and Stuart A Kauffman. 'Emergence of organisms'. In: *Entropy* 22.10 (2020), p. 1163 (cited on page 8).

[50] Maya Paczuski and Per Bak. 'Self-organization of complex systems'. In: *arXiv preprint cond-mat/9906077* (1999) (cited on page 8).

[51] Martin Niss. 'History of the Lenz–Ising Model 1950–1965: from irrelevance to relevance'. In: *Archive for history of exact sciences* 63 (2009), pp. 243–287 (cited on pages 8, 21).

[52] Reinhard Folk. 'The Survival of Ernst Ising and the Struggle to Solve His Model'. In: *Order, Disorder and Criticality: Advanced Problems of Phase Transition Theory*. World Scientific, 2023, pp. 1–77 (cited on pages 8, 21).

[53] David Perkinson. 'WebGL Sandpile Simulation'. In: *https://people.reed.edu/ davidp/* (2019) (cited on page 9).

[54] Per Bak and Michael Creutz. 'Dynamics of sand'. In: *MRS Bulletin* 16.6 (1991), pp. 17–21 (cited on page 9).

[55] Donald L Turcotte. 'Self-organized criticality'. In: *Reports on progress in physics* 62.10 (1999), p. 1377 (cited on page 9).

[56] Zeev Olami, Hans Jacob S Feder, and Kim Christensen. 'Self-organized criticality in a continuous, nonconservative cellular automaton modeling earthquakes'. In: *Physical review letters* 68.8 (1992), p. 1244 (cited on page 9).

[57] Dimitrije Marković and Claudius Gros. 'Power laws and self-organized criticality in theory and nature'. In: *Physics Reports* 536.2 (2014), pp. 41–74 (cited on page 9).

[58] George David Birkhoff. *Aesthetic measure*. Harvard University Press, 1933 (cited on pages 9, 10).

[59] Abraham A Moles. 'Theorie de linformation et perception esthetique'. In: (1958) (cited on pages 9, 10).

[60] L Neumann et al. 'Defining computational aesthetics'. In: *Computational aesthetics in graphics, visualization and imaging* 2005 (2005), pp. 13–18 (cited on page 9).

[61] Daniel J Graham and David J Field. 'Statistical regularities of art images and natural scenes: spectra, sparseness and nonlinearities.' In: *Spatial vision* 21 (2007) (cited on page 9).

[62] Daniel J Graham and David J Field. 'Variations in intensity statistics for representational and abstract art, and for art from the Eastern and Western hemispheres'. In: *Perception* 37.9 (2008), pp. 1341–1352 (cited on page 9).

[63] Daniel J Graham et al. 'Mapping the similarity space of paintings: image statistics and visual perception'. In: *Visual cognition* 18.4 (2010), pp. 559–573 (cited on page 9).

[64] Immanuel Kant. *Critique of judgment*. Newcomb Livraria Press, 2008 (cited on page 9).

[65] Jürgen Schmidhuber. 'Low-complexity art'. In: *Leonardo* 30.2 (1997), pp. 97–103 (cited on page 10).

[66] Max Bense. 'Aesthetica'. In: (1954) (cited on page 10).

[67] Claude Elwood Shannon. 'A mathematical theory of communication'. In: *The Bell system technical journal* 27.3 (1948), pp. 379–423 (cited on page 10).

[68] Jürgen Schmidhuber. 'Driven by compression progress: A simple principle explains essential aspects of subjective beauty, novelty, surprise, interestingness, attention, curiosity, creativity, art, science, music, jokes'. In: *Workshop on anticipatory behavior in adaptive learning systems*. Springer. 2008, pp. 48–76 (cited on page 10).

[69] Jürgen Schmidhuber. 'A formal theory of creativity to model the creation of art'. In: *Computers and creativity*. Springer, 2012, pp. 323–337 (cited on page 10).

[70] Aenne A Brielmann and Peter Dayan. 'A computational model of aesthetic value.' In: *Psychological review* 129.6 (2022), p. 1319 (cited on page 10).

[71] Kenneth O Stanley. 'Compositional pattern producing networks: A novel abstraction of development'. In: *Genetic programming and evolvable machines* 8 (2007), pp. 131–162 (cited on page 13).

[72] David B D'Ambrosio, Jason Gauci, and Kenneth O Stanley. 'HyperNEAT: The first five years'. In: *Growing Adaptive Machines: Combining Development and Learning in Artificial Neural Networks* (2014), pp. 159–185 (cited on pages 13, 14).

[73] David Ha, Andrew Dai, and Quoc V Le. 'Hypernetworks'. In: *arXiv preprint arXiv:1609.09106* (2016) (cited on pages 13, 14).

[74] Kenneth O Stanley et al. 'Designing neural networks through neuroevolution'. In: *Nature Machine Intelligence* 1.1 (2019), pp. 24–35 (cited on pages 13, 14).

[75] David Ha. 'Neurogram'. In: *blog.otoro.net* (2015) (cited on page 13).

[76] David Ha. 'Generating Large Images from Latent Vectors'. In: *blog.otoro.net* (2016) (cited on pages 13, 35).

[77] Kevin Jiang. 'Generative Art with Compositional Pattern Producing Networks and GANs'. In: *https://kwj2104.github.io/* (2018) (cited on page 13).

[78] Xavier Snelgrove and Matthew Tesfaldet. 'Interactive CPPNs in GLSL'. In: *Proceedings of the NeurIPS 2018 Workshop on Machine Learning for Creativity and Design*. Montreal, 2018 (cited on page 13).

[79] Jeff Clune and Hod Lipson. 'Evolving 3d objects with a generative encoding inspired by developmental biology'. In: *ACM SIGEVOlution* 5.4 (2011), pp. 2–12 (cited on page 14).

[80] Adam Paszke et al. 'Automatic differentiation in PyTorch'. In: (2017) (cited on page 15).

[81] Renée Panozzo Heilbronner. 'The autocorrelation function: an image processing tool for fabric analysis'. In: *Tectonophysics* 212.3-4 (1992), pp. 351–370 (cited on page 15).

[82] Donald B Percival and Andrew T Walden. *Spectral analysis for physical applications*. cambridge university press, 1993 (cited on page 15).

[83] Claire Robertson and Steven C George. 'Theory and practical recommendations for autocorrelation-based image correlation spectroscopy'. In: *Journal of biomedical optics* 17.8 (2012), pp. 080801–080801 (cited on page 15).

[84] Christopher J Thissen and Mark T Brandon. 'An autocorrelation method for three-dimensional strain analysis'. In: *Journal of Structural Geology* 81 (2015), pp. 135–154 (cited on page 15).

[85] Nigel Goldenfeld. *Lectures on phase transitions and the renormalization group*. CRC Press, 2018 (cited on page 15).

[86] Henrik Jeldtoft Jensen. 'What is critical about criticality: In praise of the correlation function'. In: *Journal of Physics: Complexity* 2.3 (2021), p. 032002 (cited on page 15).

[87] Henrik Jeldtoft Jensen. *Self-organized criticality: emergent complex behavior in physical and biological systems*. Vol. 10. Cambridge university press, 1998 (cited on page 15).

[88] Stephen G Brush. 'History of the Lenz-Ising model'. In: *Reviews of modern physics* 39.4 (1967), p. 883 (cited on page 20).

[89] Mark EJ Newman and Gerard T Barkema. *Monte Carlo methods in statistical physics*. Clarendon Press, 1999 (cited on page 20).

[90] John Von Neumann, Arthur W Burks, et al. 'Theory of self-reproducing automata'. In: *IEEE Transactions on Neural Networks* 5.1 (1966), pp. 3–14 (cited on pages 26, 52).

[91] Fa-Yueh Wu. 'The potts model'. In: *Reviews of modern physics* 54.1 (1982), p. 235 (cited on page 26).

[92] Stephen Wolfram. 'Cellular automata as models of complexity'. In: *Nature* 311.5985 (1984), pp. 419–424 (cited on page 26).

[93] Gérard Y Vichniac. 'Simulating physics with cellular automata'. In: *Physica D: Nonlinear Phenomena* 10.1-2 (1984), pp. 96–116 (cited on page 26).

[94] G Bard Ermentrout and Leah Edelstein-Keshet. 'Cellular automata approaches to biological modeling'. In: *Journal of theoretical Biology* 160.1 (1993), pp. 97–133 (cited on page 26).

[95] Yang Li et al. 'A review of cellular automata models for crowd evacuation'. In: *Physica A: Statistical Mechanics and its Applications* 526 (2019), p. 120752 (cited on page 26).

[96] Inés Santé et al. 'Cellular automata models for the simulation of real-world urban processes: A review and analysis'. In: *Landscape and urban planning* 96.2 (2010), pp. 108–122 (cited on page 26).

[97] Eric W Weisstein. 'Elementary cellular automaton'. In: *https://mathworld. wolfram. com/* (2002) (cited on page 27).

[98] Martin Gardner. 'The Fantastic Combinations of Jhon Conway's New Solitaire Game'Life'. In: *Sc. Am.* 223 (1970), pp. 20–123 (cited on page 26).

[99] Stephen Wolfram. *A New Kind of Science*. English. Wolfram Media, 2002 (cited on page 28).

[100] *Online database of objects in Conway's Game of Life*. URL: `https://catagolue.hatsya.com/home`. (accessed: 24.05.2024) (cited on page 28).

[101] Paul Rendell. *Turing Machine implemented in Conway's Game of Life*. 2000. URL: `http://rendell-attic.org/gol/tm.htm`. (accessed: 24.05.2024) (cited on page 28).

[102] Nicolas Loizeau. *A computer in Conway's Game of Life*. 2002. URL: `https://www.nicolasloizeau.com/gol-computer`. (accessed: 24.05.2024) (cited on page 28).

[103] Paul Rendell. 'Turing universality of the game of life'. In: *Collision-based computing*. Springer, 2002, pp. 513–539 (cited on page 28).

[104] Phillip Bradbury. *Life in Life*. 2012. URL: `https://github.com/mrphlip/life3/tree/main/life2`. (accessed: 24.05.2024) (cited on page 28).

[105] William Gilpin. 'Cellular automata as convolutional neural networks'. In: *Physical Review E* 100.3 (2019), p. 032402 (cited on page 28).

[106] Alexander Mordvintsev et al. *Growing Neural Cellular Automata*. 2020. DOI: `10.23915/distill.00023`. URL: `https://distill.pub/2020/growing-ca` (cited on page 29).

[107] Eyvind Niklasson et al. *Self-Organising Textures*. 2021. DOI: `10.23915/distill.00027.003`. URL: `https://distill.pub/selforg/2021/textures` (cited on page 29).

[108] Ettore Randazzo et al. *Self-classifying MNIST Digits*. 2020. DOI: `10.23915/distill.00027.002`. URL: `https://distill.pub/2020/selforg/mnist` (cited on page 29).

[109] Moshe Abeles. *Corticonics: Neural circuits of the cerebral cortex*. Cambridge University Press, 1991 (cited on page 31).

[110] Larry F Abbott and Sacha B Nelson. 'Synaptic plasticity: taming the beast'. In: *Nature neuroscience* 3.11 (2000), pp. 1178–1183 (cited on page 31).

[111] Robert S Zucker and Wade G Regehr. 'Short-term synaptic plasticity'. In: *Annual review of physiology* 64.1 (2002), pp. 355–405 (cited on page 31).

[112] Jesper Sjöström, Wulfram Gerstner, et al. 'Spike-timing dependent plasticity'. In: *Spike-timing dependent plasticity* 35 (2010) (cited on page 31).

[113] Natalia Caporale and Yang Dan. 'Spike timing–dependent plasticity: a Hebbian learning rule'. In: *Annu. Rev. Neurosci.* 31 (2008), pp. 25–46 (cited on page 31).

[114] P Jesper Sjostrom et al. 'Dendritic excitability and synaptic plasticity'. In: *Physiological reviews* 88.2 (2008), pp. 769–840 (cited on page 31).

[115] Anna Levina, J Michael Herrmann, and Theo Geisel. 'Dynamical synapses causing self-organized criticality in neural networks'. In: *Nature physics* 3.12 (2007), pp. 857–860 (cited on page 31).

[116] Tanguy Fardet and Anna Levina. 'Simple models including energy and spike constraints reproduce complex activity patterns and metabolic disruptions'. In: *PLoS Computational Biology* 16.12 (2020), e1008503 (cited on page 31).

[117] Ami Citri and Robert C Malenka. 'Synaptic plasticity: multiple forms, functions, and mechanisms'. In: *Neuropsychopharmacology* 33.1 (2008), pp. 18–41 (cited on page 31).

[118] Lucilla De Arcangelis, Carla Perrone-Capano, and Hans J Herrmann. 'Self-organized criticality model for brain plasticity'. In: *Physical review letters* 96.2 (2006), p. 028107 (cited on page 31).

[119] Roxana Zeraati, Viola Priesemann, and Anna Levina. 'Self-organization toward criticality by synaptic plasticity'. In: *Frontiers in Physics* 9 (2021), p. 619661 (cited on page 31).

[120] Juan A Bonachela and Miguel A Munoz. 'Self-organization without conservation: true or just apparent scale-invariance?' In: *Journal of Statistical Mechanics: Theory and Experiment* 2009.09 (2009), P09009 (cited on page 31).

[121] Juan A Bonachela et al. 'Self-organization without conservation: are neuronal avalanches generically critical?' In: *Journal of Statistical Mechanics: Theory and Experiment* 2010.02 (2010), P02015 (cited on page 31).

[122] Yoshua Bengio, Aaron C Courville, and Pascal Vincent. 'Unsupervised feature learning and deep learning: A review and new perspectives'. In: *CoRR, abs/1206.5538* 1.2665 (2012), p. 2012 (cited on page 31).

[123] 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 page 31).

[124] 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 31).

[125] Ralf Der and Georg Martius. *The playful machine: theoretical foundation and practical realization of self-organizing robots*. Vol. 15. Springer Science & Business Media, 2012 (cited on page 31).

[126] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology press, 2005 (cited on page 31).

[127] Siegrid Löwel and Wolf Singer. 'Selection of intrinsic horizontal connections in the visual cortex by correlated neuronal activity'. In: *Science* 255.5041 (1992), pp. 209–212 (cited on page 31).

[128] Patricio Gonzalez Vivo and Jen Lowe. 'The Book of Shaders'. In: *https://thebookofshaders.com/* (2015) (cited on page 35).

[129] Ed Pizzi. 'CPPN AI art project'. In: *https://edpizzi.com/* (2018) (cited on page 35).

[130] Alexander Mordvintsev et al. *Differentiable Image Parameterizations*. 2018. DOI: `10.23915/distill.00012`. URL: `https://distill.pub/2018/differentiable-parameterizations` (cited on page 35).

[131] Xavier Snelgrove and Matthew Tesfaldet. 'Interactive CPPNs in GLSL'. In: *NeurIPS Workshops*. Vol. 1. 2018 (cited on page 35).

[132] Sina Khajehabdollahi and Olaf Witkowski. 'Evolution towards criticality in Ising neural agents'. In: *Artificial Life* 26.1 (2020), pp. 112–129 (cited on page 39).

[133] Jan Prosi et al. 'The dynamical regime and its importance for evolvability, task performance and generalization'. In: *ALIFE 2021: The 2021 Conference on Artificial Life*. MIT Press. 2021 (cited on page 39).

[134] Poornima Ramesh et al. 'Indistinguishable network dynamics can emerge from unalike plasticity rules'. In: *bioRxiv* (2023), pp. 2023–11 (cited on page 42).

[135] Dorit Baras and Ron Meir. 'Reinforcement learning, spike-time-dependent plasticity, and the BCM rule'. In: *Neural Computation* 19.8 (2007), pp. 2245–2279 (cited on page 42).

[136] Robert Legenstein, Dejan Pecevski, and Wolfgang Maass. 'A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback'. In: *PLoS computational biology* 4.10 (2008), e1000180 (cited on page 42).

[137] Răzvan V Florian. 'Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity'. In: *Neural computation* 19.6 (2007), pp. 1468–1502 (cited on page 42).

[138] Corentin Tallec and Yann Ollivier. 'Can recurrent neural networks warp time?' In: *arXiv preprint arXiv:1804.11188* (2018) (cited on page 46).

[139] Silvan C Quax, Michele D'Asaro, and Marcel AJ van Gerven. 'Adaptive time scales in recurrent neural networks'. In: *Scientific reports* 10.1 (2020), p. 11360 (cited on page 46).

[140] Nicolas Perez-Nieves et al. 'Neural heterogeneity promotes robust learning'. In: *Nature communications* 12.1 (2021), p. 5791 (cited on page 46).

[141] Nicholas Soures and Dhireesha Kudithipudi. 'Deep liquid state machines with neural plasticity for video activity recognition'. In: *Frontiers in neuroscience* 13 (2019), p. 457929 (cited on page 53).

[142] Amirhossein Tavanaei et al. 'Deep learning in spiking neural networks'. In: *Neural networks* 111 (2019), pp. 47–63 (cited on page 53).

[143] Jens Wilting and Viola Priesemann. 'Inferring collective dynamical states from widely unobserved systems'. In: *Nature communications* 9.1 (2018), p. 2325 (cited on page 54).

[144] Viola Priesemann, Anna Levina, and Jens Wilting. 'Assessing criticality in experiments'. In: *The functional role of critical dynamics in neural systems* (2019), pp. 199–232 (cited on page 54).