Probing the Inductive Bias of Neural Networks through Learning Random Cellular Automata

Anonymous Author(s)
Affiliation
Address
email

Abstract

In this paper, we empirically examine whether the inductive bias of deep networks can be linked to structural properties of dynamical systems inspired by physics, such as symmetry, locality, and coarse-grained observation of outcomes. To explore this question, we generate "toy universes" by sampling random cellular-automaton rules that satisfy these constraints, and train convolutional neural networks (CNNs) to predict their evolution under three experimental factors: temporal coarse-graining, spatial pooling, and a structured (low-entropy) initial state. Throughout, we measure each network's average generalization performance relative to a baseline. While classical constraints such as symmetry and locality are necessary, they alone are not sufficient for learnability. However, when we account for the perturbation sensitivity of the target function, we observe a strong negative correlation with learnability. Further, using a structured (low-entropy) initial state leads networks to favor coarser macroscopic patterns over details.

1 Introduction

In recent years, deep learning has been the driving force that has for the first time allowed applications of machine learning to complex, "real-world" problems. From a conceptual perspective, not only is the remarkable generalization performance on complex tasks intriguing; the fact that the same class of techniques worked on a large variety of problems and data, from images and music to language and quantum chemistry with only minor and generic domain-specific adaptations [17], might be equally surprising. As statistical learning requires a strong inductive bias to generalize [44, 10, 16], with a gap exponential in input dimensionality [26], this implies that all these different kinds of data must share a common and, mathematically speaking, highly specific statistical structure. From this perspective, understanding the *prior of deep learning* relates to finding common structure in (most) naturally forming patterns. Once the existence of a common inductive bias in deep learning became apparent, identifying the principles behind it – in other words, characterizing the "prior of deep learning" – rose to a major scientific question. And it suggests a compelling hypothesis: Do the laws of physics already bias pattern formation in a way that enables intelligence as such, and deep learning in particular? More specifically, might some of the *structural principles* that physics itself follows [26] be responsible for generalizable pattern recognition? These principles include e.g. spatio-temporal *symmetry*, *locality* or entropy increasing with time (e.g. as reversible evolution starting in low-entropy initial conditions). Unsurprisingly, finding hard evidence for such a link has proven to be non-trivial. In this paper, we try to approach the question of linking these structural principles to learnability by deep neural networks through an experimental approach: We fix some of the structural principles and then build a randomly chosen "toy universe" that follows those and try to learn to predict patterns in the resulting system via deep learning. This begs the question of which superset of "universes" to and how to define "learning the patterns". At this point, we have to make rather strong simplifying assumptions:

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

Laws & Universes: In order to make a study tractable, we have to restrict ourselves to a discrete set of candidate laws, and we have chosen cellular automata (CAs) with binary cells as this could be considered a minimal model system. Obviously, this serves as an analogon for a qualitative understanding, not as a model to describe actual real-world physics.

Learning Patterns: Similarly aiming at a simple formalization of "learning" pattern formation, we pick the task of simply predicting the future evolution of given initial conditions. We assume perfect knowledge of the initial state (input) but study various degrees of temporal and spatial coarse-graining (on the output side) to study scale effects.

We also have to be careful in evaluation and interpretation of the results:

Performance: For evaluation, we measure generalization performance, not memorization. To quantify success, we relate performance to a simple baseline to exclude being misled by a trivial success of rules that do not generate meaningful patterns, the increased frequency of which can be a side effect of constraining the dynamics. We also make sure to train in a regime that would permit identifying the (purely randomly chosen) rules when the additional coarse-graining structure was fully known and modeled by the learning system (which it is not), as it is obviously not possible to guess random bits of information with any conceivable piece of prior information.

Level of Emergence: Our experiments use only moderate numbers of timesteps and moderate coarse-graining, i.e., they involve only a small number of computational steps. Thus, they could be interpreted as an analogon of a rather simple, macroscopic physical system with a small number of interactions and interacting parts. Describing structure formation along large scale differences and time scales would bring up insurmountable problems of undecidability and lack of statistical power.

Despite requiring fundamental computational compromises, our study reveals some interesting insights: First, we can easily convince ourselves that *locality and symmetry are necessary* in the setting outlined above, but, empirically, turn out *not sufficient*. Including *sensitivity* of the overall computation to perturbations of initial conditions as a parameter shows a significant correlation with training success, declining with sensitivity and timesteps, but still not being sufficient. Further, we investigate the impact of structured "low-entropy" initial conditions: Here the network is able to predict the coarse-scale "shape" of the output - corresponding to the spread of information in the automaton, but is often unable to predict the fine-scale " texture", suggesting that neural networks are able to fit low information subsets of a problem when possible, while ignoring more complex parts.

Overall, by using a novel experimental approach of building "random toy universes" under constraints on its "physics", we can show that the combination of principles examined is strongly correlated, but not fully sufficient for learnability. The persistent simplicity bias could point towards a hidden, yet unknown principle of favoring stability in structure formation.

2 Related Work

Universal Priors: The paradox of universal induction has intrigued researchers for a long time, with the impossibility result of "no-free-lunch" [44, 16] generally resolved by an appeal to a variant of Occam's razor [37, 38, 35, 27, 19], which demands concise coding and breaks the symmetry by assuming a mathematical language. This does not relate to natural dynamics in an obvious way [42].

Priors of Deep Learning: Prior knowledge can be incorporated explicitly, e.g., by exploiting symmetry [14, 8] or multi-scale modeling (e.g., via pooling [46]). But even for a basic fully connected MLP, biases are known. Prominently, they learn low-frequency features more quickly [34]. Using a tangent-linear model (NTK) of network training [31, 20], this bias can be understood as decaying eigenvalues of kernel-eigenfunctions [7]. This tangent-spectral picture also explains phenomena such as double descent [4, 3], grokking [24] and, to some extent, adversarial examples [40]. The notion of *sensitivity* (susceptibility of a function to small input perturbations [21]) is closely related to the spectral bias [41]: A correlation between discrete sensitivity and generalization in neural networks has been established early on [12], and similar findings hold in a large empirical study for various notions of sensitivity (including continuous ones, such as sharp minima) [32]. The findings have been replicated for recurrent architectures and transformers, which showed an even stronger bias [6]. Again, an NTK model provides an explanation [41] with an explicit link between continuous low-frequency bias and discrete sensitivity, as we use in our paper. Given what we already know

about sensitivity, the main insight in our paper is that sensitivity seems to be orthogonal to the other structural principles of physics and highly correlated with, but not sufficient for generalization.

In a different line of work, Mingard et al. [28] propose that the prior of deep networks can be characterized as simple Bayesian sampling from the initialization distribution in function space, with good empirical matches in a Gaussian approximation. They also show that this induces a low-complexity prior [29]. While constituting a big step towards a better understanding of the inductive bias of deep learning, it does not address the question of a connection to dynamical processes.

Links to Physical Dynamics: Machine learning as a field has been influenced strongly by concepts from natural science and physics [36, 47]. In terms of causal links far-reaching hypotheses have been proposed, such as linking predicting dynamics to self-preserving intelligent structures [13], or dualities of learning and fundamental physics [2]; however, it remains challenging to prove or disprove models at this scope. The influential article by Lin et al. [26] enumerates concrete links between physical models and structural properties of deep networks at a formal level, in particular showing their ability to encode low-order polynomial Hamiltonians, exploiting symmetry for compactness, and re-addressing renormalization (also addressed elsewhere, e.g. [9]) for bridging scales, but still leaves open how complex emergent structures are captured.

Cellular automata (CAs) have long been used as model systems for physical dynamics, both in a concrete sense of discretization of fundamental physics [18] as well as merely an abstract analogon, as in our paper. Connections between sensitivity and complexity measures such as entropy and Lyapunov exponents have been studied by Langton [25], referring to Wolfram's foundational categorization [43]. CAs have also already been used as model system in studying neural networks: Wulff and Hertz [45] learn single timesteps and already identify that chaotic CA rules posed significant learning challenges. Gilpin [15] demonstrate theoretically and empirically that CNN architectures are capable of explicitly encoding the local rules underlying CA dynamics, asserting that given sufficient training data, CNNs can precisely replicate CA update rules. Springer and Kenyon [39] show that the Turing-complete Game-of-Life is hard to learn in a setting of temporal coarse-graining. Elser [11] follow-up by design a training protocol to sample good training examples. Aach et al. [1] explored generalization across multiple CA rules, finding that CNNs could partially generalize to unseen configurations and even unseen rules within certain constraints. On the flip-side, Neural Cellular Automata [30] demonstrate that strictly local iterative rule application can be learned that results in highly complex patterns. Bhamidipaty et al. [5] use model systems for algorithm evaluation, including CAs, but their work does not aim at links to physics. Our study differs to previous work in its approach to study the connection between physically-motivated constraints and learnability.

3 Methods

3.1 Cellular Automata as Discrete Dynamical Systems

We model discrete dynamical systems using Cellular Automata (CA). A CA describes the evolution of a state defined over a discrete grid and time. Formally, let the state be a function $s : \Omega \times \mathbb{Z} \to \mathbb{B}$, where $\Omega \subset \mathbb{Z}^d$ is the spatial grid (here $d = 2$), $t \in \mathbb{Z}$ is discrete time, and $\mathbb{B} := \{0, 1\}$ represents the binary state of each cell. The evolution is governed by a local transition function $f : \mathbb{B}^k \to \mathbb{B}$ applied synchronously to all cells:

$$s(r, t + 1) = f(\mathcal{N}(r, t)) \tag{1}$$

where the neighborhood $\mathcal{N}$ contains k cells spatially adjacent to r and r itself at time t. The function f and an initial condition $s(\,\cdot\,, t_0)$ determine the system's entire trajectory. For convenience, we also use f to denote the map $s(\,\cdot\,, t) \to s(\,\cdot\,, t + 1)$, obtained by applying f to every cell at the same time. Specifically, the state s is defined on an $H \times W$ grid with binary values ($\mathbb{B} = \{0, 1\}$) and torus topology (periodic boundary conditions). We use a $k = 9$ Moore neighborhood (the 3×3 square centered on the cell r). The transition function $f : \mathbb{B}^9 \to \mathbb{B}$ maps each of the $2^9 = 512$ possible neighborhood states to a next state for the central cell. We mostly use outer-totalistic CAs, where only the sum of the neighborhood and the value of the central cell are used to calculate the next state. We do so as outer-totalistic CA have a higher chance of exhibiting interesting behavior and have a lower complexity. Nonetheless, control experiments with fully general rules did not have qualitatively different outcomes and results of these experiments are provided in the Appendix. Rules are sampled uniformly at random from the entire rule-space ($\#\mathcal{F} = 2^{18}$ for outer-totalistic automata) for each experiment, unless otherwise specified.

3.2 Coarse-Graining

To model realistic scenarios where observations of dynamical systems are typically imperfectly resolved in time and space, we introduce two coarse-graining procedures: temporal and spatial. These allow us to study how varying levels of granularity influence learnability by deep neural networks.

Temporal coarse-graining is implemented by repeatedly composing the CA update function $f : \mathbb{B}^{H\times W} \to \mathbb{B}^{H\times W}$ over multiple discrete timesteps. For a given temporal scale T, we define the temporally coarse-grained mapping $f^{(T)}$ as:

$$f^{(T)}(x) = f(f(\dots f(x))) \quad (T \text{ times}). \tag{2}$$

By training neural networks directly to predict $f^{(T)}(x)$ from initial conditions x, we test the networks' ability to handle compounded dynamics and identify whether there are temporal thresholds beyond which the complexity becomes unlearnable.

Spatial coarse-graining reduces the spatial resolution of the CA state after evolution. In our experiments, we implement spatial coarse-graining through a majority-voting pooling operator P_S, which aggregates each overlapping $S \times S$ neighborhood of cells into a single binary value. Formally, spatial coarse-graining transforms the evolved CA state as follows:

$$x \mapsto P_S(f^{(T)}(x)). \tag{3}$$

Spatial coarse graining accounts for observations at limited resolution. By varying the spatial pooling factor S, we can study how spatial resolution influences neural network learnability (positively or negatively). Together, these two coarse-graining procedures allow us to characterize how neural network learnability depends on the level of detail available in both temporal and spatial domains.

3.3 Fundamental Constraints: Locality and Symmetry

Our choice of CAs implicitly incorporates structural constraints analogous to those in physical systems, namely locality, symmetry, and complexity. We discuss each below:

Locality: The update rule f depends only on a small, spatially contiguous neighborhood ($k = 9$). This restriction is crucial from an information-theoretic perspective, as in the fully general case, the number of possible transition functions grows doubly-exponentially by 2^{2^k} with the neighborhood size k [43]. Less locality would thus require an exponentially larger amount of training data to identify. Our CNN architecture (Sect. 3.5) incorporates this prior explicitly.

Symmetry (Spatial and Temporal Invariance): The same transition rule f is applied identically at all spatial locations r and all timesteps t. This spatio-temporal symmetry is essential for generalization in our setting of CAs. Generally speaking, invariance forms the foundation of inductive reasoning as it permits reproducible experiments (or identically distributed training data, in statistical terms). Our CNN models spatial symmetry over the full receptive field (growing with temporal coarse-graining) but does not impose temporal symmetry within the network (i.e., weights are not shared across layers). The 90° rotational/reflective symmetry of outer-totalistic CAs is not exploitet.

Complexity: Empirical studies of neural networks demonstrate a strong inductive preference towards simpler functions, often described as a simplicity bias related to Kolmogorov complexity $K(f)$ [29]. Within our experimental framework, the complexity of the learned functions — specifically, the mapping from initial states to evolved states — is naturally constrained by the simplicity of the underlying CA transition rule. As a result, the functions we attempt to learn have significantly lower complexity than arbitrary binary functions defined on grids of comparable size. This inherent simplicity ensures that observed differences in learnability predominantly reflect meaningful structural properties rather than information-theoretic limitations.

3.4 Training Data and Initial Conditions

Training data consists of pairs (s_0, s_T) where $s_0 \in \mathbb{B}^{N\times N}$ is an initial state and $s_T = f^{(T)}(x)$ is the state after T timesteps under the chosen rule f. We generate initial conditions s_0 using three distinct procedures to probe different aspects of network learnability:

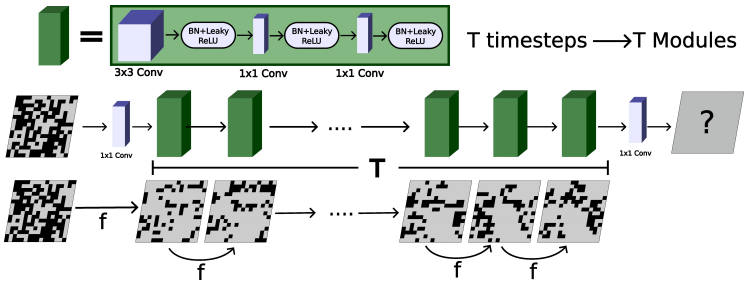

Figure 1: Schematic overview: The CNN architecture used consists of T blocks, the same amount as timesteps to be modeled. This guarantees that it is able to fit a given function at the given timescale. We use LeakyReLU and BatchNorm after each convolutional layer except for the last one.

- **Random initialization:** Each cell $s_{0_{ij}}$ is sampled independently and uniformly from $\mathbb{B}$, i.e., $P(s_{0_{ij}} = 0) = P(s_{0_{ij}} = 1) = 0.5$. This procedure serves as a standard baseline to measure generalization under uniform randomness.
- **Naturalized states:** To produce a "steady-state" configurations, we first initialize states randomly as in (1), and then evolve them by applying f for a small number of steps t. The resulting training data consists of pairs (s_t, s_{T+t}). This approach assesses the network's capacity to predict CA dynamics from states closer to their natural, evolved distributions.
- **Localized initialization:** Initial states are generated randomly as in (1), but with a fixed border of width T pixels on each side set to 0. This construction explicitly tests the influence of spatial gradients in information density on learnability.

The first two procedures correspond to constraints of maximum entropy, with the second one modeling thermal equilibrium. The third represents constraints to the initial state of lower entropy.

3.5 Network Architecture and Training

Our model architecture is a fully convolutional network designed to respect the locality and spatial invariance of the CA dynamics. To predict T steps ahead, the network consists of T sequential blocks, giving the opportunity to model one timestep per block. Crucially, these blocks do *not* share weights, allowing the network to learn potentially distinct intermediate representations at each step.

Each block consists of:

1. A 3×3 Conv2D layer, modeling interactions within the Moore neighborhood. Circular padding is used to maintain spatial dimensions and periodic boundary conditions.
2. Followed by two 1×1 Conv2D layers, increasing representational expressivity while preserving locality.
3. BatchNorm and LeakyReLU activations between each convolutional layer for stabilization and non-linearity.

The network's final prediction is obtained through a 1×1 convolutional layer outputting two channels, corresponding to binary logits for each cell. All intermediate convolutional layers employ 128 feature channels. A schematic overview is provided in Figure 1. Note that structuring the network as one block per timestep guarantees that the network is able to model the function we are trying to learn, as a single block, in principle, has the ability to model a single timestep. Networks are trained using the Adam optimizer [23] with a learning rate of 4×10^{-4} and a binary cross-entropy loss. Training is performed for 4096 iterations with a batch size of 64, using patches of size 16×16 or 32×32 cells depending on the specific experiment. This procedure makes $T = 1$ treatments (i.e., the rules as such) easily learnable, but is non-trivial for larger T (experimentally visible in Fig.2 for T=2,3,...).

Hyperparameters such as learning rate, batch size, and channel depth were selected based on smaller-scale preliminary experiments to ensure stable and effective training.

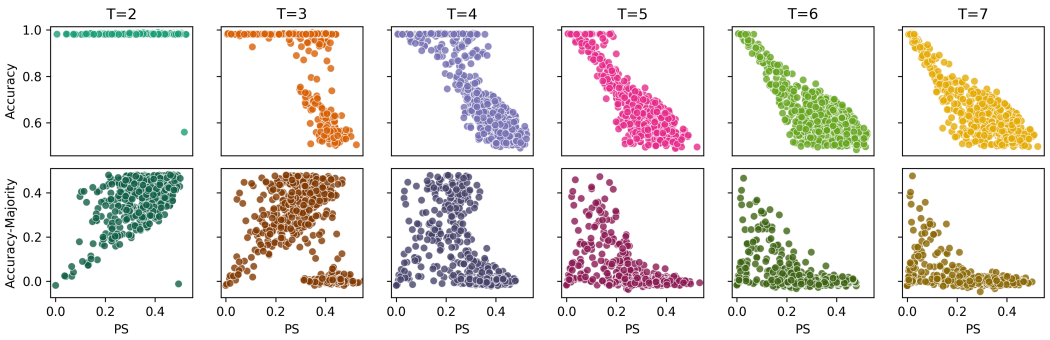

Figure 2: Results of different training runs for different timescales. From left to right, coarse-graining increases from $T = 2$ to $T = 7$. First row shows accuracy of the trained network, second run shows accuracy difference compared to a simple majority classifier. We can see that with increased temporal coarse-graining it becomes more difficult to learn functions with higher perturbation sensitivity.

3.6 Evaluation Metrics

To quantify learnability, we monitor performance metrics of the neural networks:

- **Pixel-wise Accuracy:** The fraction of correctly predicted cell states.
- **Accuracy Gain over Baselines:** We compare network accuracy to a baseline of predicting the majority class. We chose this baseline as networks frequently defaulted to constant predictions when training failed, allowing for an easy way to compare to the "failure case". In one scenario, we additionally include comparisons to a logistic regression classifier.

We also measure the **Perturbation Sensitivity (PS)** of $f^{(T)}$ as a computationally feasible proxy for the complexity of the mapping. PS quantifies how sensitive the system evolution is to small input changes by measuring the average effect of flipping a single bit in the input state, formally:

$$\mathrm{PS}(f^{(T)}) \approx \frac{1}{N}\sum_{n=1}^{N} \mathbb{E}_{i\sim\mathrm{Unif}(1,HW)}\left[\frac{1}{HW}\sum_{j=1}^{HW} |f^{(T)}(x_n)_j - f^{(T)}(x_n^{(i)})_j|\right] \tag{4}$$

where $x_n^{(i)}$ denotes the state x_n with its i-th bit flipped. Unlike alternative complexity metrics (e.g., Kolmogorov complexity, Lempel-Ziv complexity), PS is efficiently computable and directly measures the sensitivity relevant to prediction robustness [12].
We conduct a broad experimental sweep across various rules f, temporal depths T and spatial coarse-graining sizes S, examining relationships between PS and predictive accuracy for standard and naturalized initialization. We also systematically compare performance under globally randomized versus localized initializations.

4 Experiments

4.1 Learnability and Temporal coarse-graining

To investigate the effect of temporal coarse-graining on learnability, we sample 512 random outer-totalistic CA rules f. For each rule, we train separate CNNs to predict the state after T timesteps, where $T \in \{2, 3, 4, 5, 6, 7\}$ for a total of 3072 training runs, taking $\sim$15 hours on an NVIDIA GeForce RTX 4090. We measure validation accuracy achieved by the network after 4096 training iterations and compute the Perturbation Sensitivity (PS) of the corresponding T-step function $f^{(T)}$.

Figure 2 shows that predicting short time horizons ($T = 2, 3$) is feasible for most rules, with networks typically achieving high accuracy, significantly outperforming the majority-vote baseline, and often reaching near-perfect ($\sim 100\%$) prediction accuracy. However, as the prediction horizon T increases, learnability deteriorates rapidly. For $T \geq 4$, highly sensitive rules become effectively unpredictable by the CNN, with accuracy collapsing towards baseline performance. Moreover, Figure 2 illustrates

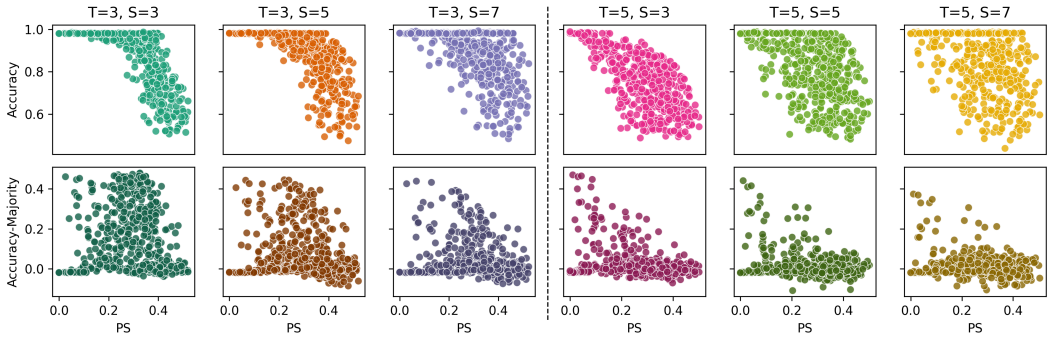

Figure 3: Results of different training runs for different spatial coarse-grainings. From left to right, coarse-graining increases from $S = 3$ to $S = 7$ with temporal coarse-graining of $T = 3$ left and $T = 5$ right. First row shows accuracy of the trained network, second run shows accuracy difference compared to a majority classifier baseline. While accuracy seems to increase with larger coarse-graining, comparison with the baseline shows that this is mainly driven by rule simplification.

a clear relationship between PS and the accuracy achieved by the CNN. Accuracy consistently decreases as PS increases, especially at higher temporal scales (T). This trend remains even when measuring accuracy relative to the baseline. Networks consistently outperforming the baseline appear predominantly at lower PS values, with the maximum PS at which CNNs outperform the baseline decreasing as the temporal scale increases. At large T, high sensitivity excludes outperforming the base-line. Low sensitivity is not sufficient but increases the chance of learnability.

We repeated these experiments using naturalized initial conditions, resulting in an additional 3072 training runs, taking an additional $\sim$ 22 hours. Results were qualitatively similar, confirming the robustness of the observed trends and the impact of perturbation sensitivity on predictability.

4.2 Learnability and Spatial coarse-graining

To evaluate how spatial coarse-graining impacts learnability, we again sample 512 random outer-totalistic CA rules. For each rule, we train networks to predict CA evolution under combinations of temporal $T = 3, 4, 5$ and spatial coarse-graining scales $S = 3, 5, 7$, resulting in a total of 4608 training runs, taking ~ 33 hours of training time. Because spatial coarse-graining effectively enlarges each cell's receptive field, we increase CNN depth by adding $1, 2$, or 3 convolutional blocks for spatial scales $S = 3, 5, 7$, respectively. Results from spatial coarse-graining experiments differ notably from those obtained with temporal coarse-graining alone, as can be seen in Figure 3. While absolute accuracy generally increases with greater spatial coarse-graining, improvements relative to the majority-vote baseline are much less pronounced. Indeed, as spatial coarse-graining increases, the baseline predictor becomes inherently stronger, as it benefits from pooling, leaving less possibilities for CNN improvement. As a result, CNNs rarely outperform the baseline for functions exhibiting high perturbation sensitivity. Still, the relationship between lower PS and improved relative performance remains apparent, but weaker than in purely temporal experiments.

Repeating the spatial coarse-graining experiments with naturalized initial conditions (another 4608 training runs, $\sim 50h$ compute time) yields largely similar results, with one notable exception: a distinct cluster of rules emerges exhibiting low PS and high accuracy relative to the baseline. Analysis indicates that this cluster consists of rules converging to fixed points, thereby simplifying the prediction task to an identity mapping. Replacing the majority-vote baseline with logistic regression removes this cluster, producing results similar to those obtained with randomly initialized states.

4.3 Learnability and Localized Initial Conditions

To examine how localized entropy influences learnability, we perform experiments comparing training outcomes under localized initial conditions (as defined in Section 3.4). As in previous sections, we sample 512 random CA rules and train CNNs to predict their states after temporal coarse-graining with horizons $T = 4, 5, 6$ for a total of 1536 runs. However, instead of initializing the entire patch uniformly at random, we use a structured initialization, fixing a border of width T pixels on each side

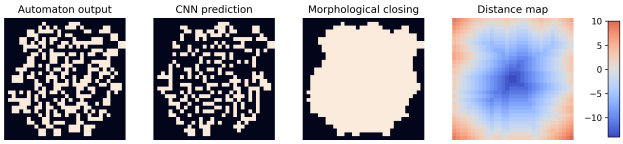

Figure 4: How distance to objects are calculated. To the left we see the automaton output and the CNN prediction. We apply a morphological close to the automaton output and calculate signed distances to the resulting object. Positive distances correspond to the exterior (red), negative ones to the interior (blue).

to zero. Because such a setup yields very small central regions on 16×16 patches, we perform the experiments on larger 32×32 patches, taking $\sim$20 hours of compute time.

While the overall prediction accuracy does not clearly improve under these localized initializations, examining predictions visually reveals an interesting structural phenomenon. The network is often able to predict the overall "shape" of the evolved pattern correctly, achieving significantly higher accuracy along the edges of these shapes compared to their interiors. While improved accuracy on the outermost edges of the 32×32 patch might be trivially expected, the phenomenon is present even for smaller shapes formed by localized initializations.

To quantitatively evaluate this observation, we analyze prediction accuracy relative to spatial distance from the shape boundary. Specifically, given a prediction and a correct label, we first run a morphological closing operation on the label. We then calculate the signed l_1 distance from each cell to this shape, assigning positive distances to cells outside the shape and negative distances to cells inside (see Figure 4 for an illustration). For each prediction-label pair, we calculate average accuracy for each distance, and aggregate results across 32 independent prediction-label instances per rule.

Examples (see example visualizations in Figure 6, and aggregated results over 100 randomly selected rules in Figure 5) confirm the observed trend. CNN predictions consistently approach near-perfect accuracy ($\sim 100\%$) at positive distances (outside the predicted shape) but show a rapid and consistent decline at negative distances (inside the shape). A large fraction of prediction accuracy thus stems from the CNN's ability to infer how far spatial information can propagate from the local initialization, though the exact pattern details remain difficult for the network to predict accurately. Consequently, the CNN reliably matches the overall output shape—and occasionally simpler interior details, but struggles with complex fine-grained internal structure.

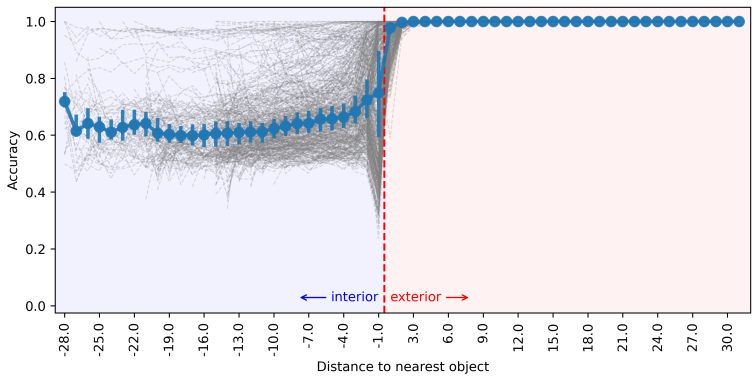

Figure 5: Accuracy by distance for 400 randomly drawn rules. Each gray line corresponds to the mean accuracy for the given distance in one such example, calculated over a batch of 32 samples. Blue dots correspond to the median accuracy over all rules, with blue bars representing a 25% quartile around the median. The light-blue region corresponds to the interior, the light-red to the exterior. We can see that on average a CNN loses substantial amount of accuracy on the edge of the automaton, and then continues to lose accuracy further in the interior. In the innermost parts, the models regain accuracy, but this is partially an artifact of few examples with large negative distances.

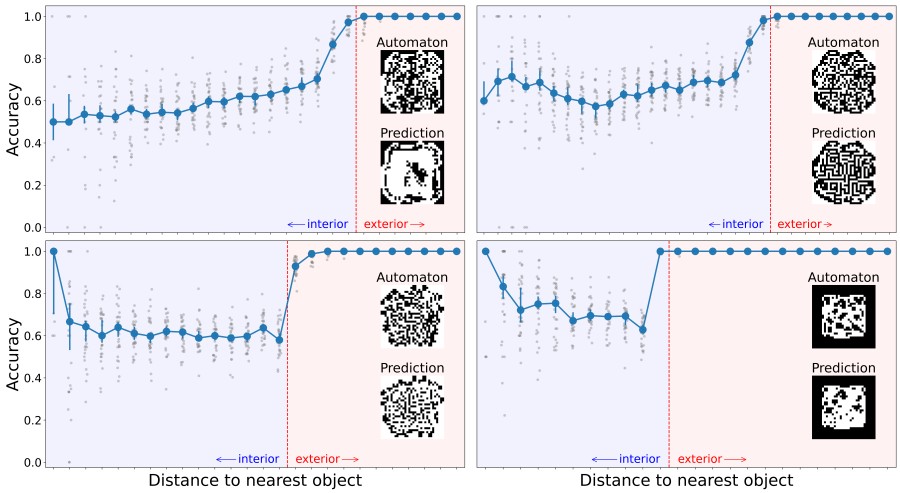

Figure 6: Accuracy by distance to closure over a batch of 32 examples for 4 different automata. Each blue dot corresponds to the median accuracy in one such example, while the blue bars represent the surrounding 25% quartile. We also show a single output of the automaton and the corresponding network prediction. While the network is able to predict the outer edges of the automaton, it is unable to infer the interior with the same accuracy. More examples are provided in the Appendix.

5 Discussion and Conclusions

Our experiments using CAs as model systems provide several insights into the conditions influencing DNN learnability of dynamical systems. First, fundamental physical constraints like locality and spatio-temporal symmetry, while necessary and naturally embodied by both CAs and CNNs, are insufficient to guarantee predictability. CNNs often failed to learn the evolution of even simple, local, and symmetric CA rules beyond short time horizons ($T \approx 3$–4) when trained on high-dimensional, randomly initialized inputs. Second, system sensitivity, measured by PS, correlates with learnability in a scale-dependent way. At longer time horizons, high sensitivity leads to more frequent prediction collapse. However, this relationship is not linear or complete, suggesting that other aspects of system complexity, such as long-range correlations, emergent macrostructures, or proximity to criticality (e.g., Wolfram's classification [43] or Langton's λ parameter [25]), may also shape learnability boundaries. Third, the structure of the initial state has a strong impact on learnability. We see clear differences in accuracy along the "edges" of predicted structures under localized initial conditions, which suggests that predictive failure in the globally random case is due to the difficulty of tracking numerous signals across the entire receptive field. This implies that not just the rule's complexity, but also the distribution and organization of the input state, play a role in determining learnability. These findings are consistent with challenges observed in other domains, such as long-term prediction in chaotic systems [33], and the success of methods focusing on localized or sparse interactions.

Limitations of this work include the focus on 2D binary CAs, which are a simplification of continuous physical systems, and the specific CNN architecture used. Future work could explore different CA types (e.g., continuous-valued, higher dimensions), alternative complexity metrics, and architectures like Recurrent Neural Networks (RNNs) or Transformers adapted for spatio-temporal data. Our setting also does not allow the exploration of some further structural principles of classical physics, such as reversability (as testing whether a random CA is reversible is undecidable [22]) or conservation laws (which do not have a straightforward analogon in the discrete case).

In conclusion, this study underscores that DNN generalization for dynamical systems depends on the interplay between the system's rules (locality, symmetry, intrinsic complexity), the structure of the states encountered (localized vs. global activity), and the network's architectural priors. Simply matching basic symmetries like locality is not enough. The effective complexity presented by the data itself is a key determinant of learnability. While sensitivity is a known proxy highly correlated with learnability, it is not sufficient, suggesting that we are still missing a more specific measure.

References

[1] Marcel Aach, Jens Henrik Goebbert, and Jenia Jitsev. Generalization over different cellular automata rules learned by a deep feed-forward neural network, 2021. URL https://arxiv.org/abs/2103.14886.

[2] Stephon Alexander, William J. Cunningham, Jaron Lanier, Lee Smolin, Stefan Stanojevic, Michael W. Toomey, and Dave Wecker. The Autodidactic Universe. 3 2021.

[3] Mikhail Belkin. Fit without fear: remarkable mathematical phenomena of deep learning through the prism of interpolation. *Acta Numerica*, 30:203–248, 2021. doi: 10.1017/S0962492921000039.

[4] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019. doi: 10.1073/pnas.1903070116. URL https://www.pnas.org/doi/abs/10.1073/pnas.1903070116.

[5] Logan M Bhamidipaty, Tommy Bruzzese, Caryn Tran, Rami Ratl Mrad, and Maxinder S. Kanwal. Dynadojo: An extensible platform for benchmarking scaling in dynamical system identification. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 15519–15530. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/32093649cbbcff773d9a991d8c30a7fe-Paper-Datasets_and_Benchmarks.pdf.

[6] Satwik Bhattamishra, Arkil Patel, Varun Kanade, and Phil Blunsom. Simplicity bias in transformers and their ability to learn sparse Boolean functions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5767–5791, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.317. URL https://aclanthology.org/2023.acl-long.317/.

[7] Yuan Cao, Zhiying Fang, Yue Wu, Ding-Xuan Zhou, and Quanquan Gu. Towards understanding the spectral bias of deep learning. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 2205–2211. International Joint Conferences on Artificial Intelligence Organization, 8 2021. doi: 10.24963/ijcai.2021/304. URL https://doi.org/10.24963/ijcai.2021/304.

[8] Taco Cohen and Max Welling. Group equivariant convolutional networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2990–2999, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings.mlr.press/v48/cohenc16.html.

[9] Ellen De Mello Koch, Robert De Mello Koch, and Ling Cheng. Is deep learning a renormalization group flow? *IEEE Access*, 8:106487–106505, 2020. doi: 10.1109/ACCESS.2020.3000901.

[10] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2 edition, November 2000. ISBN 0471056693.

[11] Veit Elser. Reconstructing cellular automata rules from observations at nonconsecutive times. *Physical Review E*, 104(3):034301, 2021.

[12] Leonardo Franco. Generalization ability of boolean functions implemented in feedforward neural networks. *Neurocomputing*, 70(1):351–361, 2006. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2006.01.025. URL https://www.sciencedirect.com/science/article/pii/S0925231206000361. Neural Networks.

[13] Karl Friston. The free-energy principle: a unified brain theory? *Nature reviews neuroscience*, 11(2):127–138, 2010.

[14] Kunihiko Fukushima. Visual feature extraction by a multilayered network of analog threshold elements. *IEEE Transactions on Systems Science and Cybernetics*, 5(4):322–333, 1969. doi: 10.1109/TSSC.1969.300225.

[15] William Gilpin. Cellular automata as convolutional neural networks. *Physical Review E*, 100 (3), September 2019. ISSN 2470-0053. doi: 10.1103/physreve.100.032402. URL http://dx.doi.org/10.1103/PhysRevE.100.032402.

[16] Micah Goldblum, Marc Anton Finzi, Keefer Rowan, and Andrew Gordon Wilson. The no free lunch theorem, kolmogorov complexity, and the role of inductive biases in machine learning, 2024. URL https://openreview.net/forum?id=X7nz6ljg9Y.

[17] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[18] Gerard T. Hooft. *The Cellular Automaton Interpretation of Quantum Mechanics*. Springer, 2016.

[19] Marcus Hutter. *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*. Springer, Berlin, 2005. ISBN 3-540-22139-5. doi: 10.1007/b138233.

[20] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[21] Jeff Kahn, Gil Kalai, and Nathan Linial. The influence of variables on boolean functions. In *29th Symposium on the Foundations of Computer Science*, pages 68–80, White Plains, 1988.

[22] Jarkko Kari. Reversibility of 2d cellular automata is undecidable. *Physica D: Nonlinear Phenomena*, 45(1):379–385, 1990. ISSN 0167-2789. doi: https://doi.org/10.1016/0167-2789(90)90195-U. URL https://www.sciencedirect.com/science/article/pii/016727899090195U.

[23] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

[24] Tanishq Kumar, Blake Bordelon, Samuel J. Gershman, and Cengiz Pehlevan. Grokking as the transition from lazy to rich training dynamics. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=vt5mnLVIVo.

[25] Chris G. Langton. Computation at the edge of chaos: Phase transitions and emergent computation. *Physica D: Nonlinear Phenomena*, 42(1):12–37, 1990. ISSN 0167-2789. doi: https://doi.org/10.1016/0167-2789(90)90064-V. URL https://www.sciencedirect.com/science/article/pii/016727899090064V.

[26] Henry W Lin, Max Tegmark, and David Rolnick. Why does deep and cheap learning work so well? *Journal of Statistical Physics*, 168:1223–1247, 2017.

[27] David MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2004.

[28] Chris Mingard, Guillermo Valle-Pérez, Joar Skalse, and Ard A. Louis. Is sgd a bayesian sampler? well, almost. *Journal of Machine Learning Research*, 22(79):1–64, 2021. URL http://jmlr.org/papers/v22/20-676.html.

[29] Chris Mingard, Henry Rees, Guillermo Valle-Pérez, and Ard A Louis. Deep neural networks have an inbuilt occam's razor. *Nat. Commun.*, 16(1):220, January 2025.

[30] Alexander Mordvintsev, Ettore Randazzo, Eyvind Niklasson, and Michael Levin. Growing neural cellular automata. *Distill*, 2020. doi: 10.23915/distill.00023. https://distill.pub/2020/growing-ca.

[31] Radford M. Neal. *Priors for Infinite Networks*, pages 29–53. Springer New York, New York, NY, 1996. ISBN 978-1-4612-0745-0. doi: 10.1007/978-1-4612-0745-0_2. URL https://doi.org/10.1007/978-1-4612-0745-0_2.

[32] Roman Novak, Yasaman Bahri, Daniel A. Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=HJC2SzZCW.

[33] Jaideep Pathak, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott. Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach. *Phys. Rev. Lett.*, 120:024102, Jan 2018. doi: 10.1103/PhysRevLett.120.024102. URL https://link.aps.org/doi/10.1103/PhysRevLett.120.024102.

[34] Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *International Conference on Machine Learning*, pages 5301–5310. PMLR, 2019.

[35] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978. ISSN 0005-1098. doi: https://doi.org/10.1016/0005-1098(78)90005-5. URL https://www.sciencedirect.com/science/article/pii/0005109878900055.

[36] Daniel A. Roberts, Sho Yaida, and Boris Hanin. *The Principles of Deep Learning Theory: An Effective Theory Approach to Understanding Neural Networks*. Cambridge University Press, 2022.

[37] Ray Solomonoff. A formal theory of inductive inference part i. *Information and Control*, 7: 1–22, 1964.

[38] Ray Solomonoff. A formal theory of inductive inference part ii. *Information and Control*, 7: 224–254, 1964.

[39] Jacob M. Springer and Garrett T. Kenyon. It's hard for neural networks to learn the game of life. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2021. doi: 10.1109/IJCNN52387.2021.9534060.

[40] Nikolaos Tsilivis and Julia Kempe. What can the neural tangent kernel tell us about adversarial robustness? In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 18116–18130. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/72f9c316440c384a95c88022fd78f066-Paper-Conference.pdf.

[41] Bhavya Vasudeva, Deqing Fu, Tianyi Zhou, Elliott Kau, Youqi Huang, and Vatsal Sharan. Transformers learn low sensitivity functions: Investigations and implications. In *ICLR2025, The Thirteenth International Conference on Learning Representations*, 2025.

[42] Eugene P. Wigner. The Unreasonable Effectiveness of Mathematics in the Natural Sciences. *Communications in Pure Applied Mathematics*, 13(1):1–14, February 1960. doi: 10.1002/cpa.3160130102.

[43] Stephen Wolfram. Universality and complexity in cellular automata. *Physica D: Nonlinear Phenomena*, 10(1):1–35, 1984. ISSN 0167-2789. doi: https://doi.org/10.1016/0167-2789(84)90245-8. URL https://www.sciencedirect.com/science/article/pii/0167278984902458.

[44] David Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, 8:1341–1390, 10 1996.

[45] N. Wulff and J A Hertz. Learning cellular automaton dynamics with neural networks. In S. Hanson, J. Cowan, and C. Giles, editors, *Advances in Neural Information Processing Systems*, volume 5. Morgan-Kaufmann, 1992. URL https://proceedings.neurips.cc/paper_files/paper/1992/file/d6c651ddcd97183b2e40bc464231c962-Paper.pdf.

[46] Kouichi Yamaguchi, Kenji Sakamoto, Toshio Akabane, and Yoshiji Fujimoto. A neural network for speaker-independent isolated word recognition. In *First International Conference on Spoken Language Processing (ICSLP)*, pages 1077–1080, 1990.

[47] Lenka Zdeborová. Understanding deep learning is also a job for physicists. *Nat. Phys.*, 16: 602–604, 2020.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Section 4

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 5

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Architecture, training params and data are explained in Section 3 and Section 4

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification:

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Percentile Intervals are provided for Section 4.3

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Runtimes are given in Section 4, all experiments use NVIDIA 4090 GPU

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: No real world dataset, theoretical work, no clear application for unintended uses.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.