Winfree Oscillatory Neural Network

Jiawen Dai[1,2] and **Yue Song**[†1]

[1]College of AI, Tsinghua University
[2]Shanghai Jiao Tong University

Abstract

Oscillations and synchronization are widely believed to play a fundamental role in representation and computation. However, existing machine learning approaches based on synchronization dynamics have largely been confined to specialized settings such as object discovery, with limited evidence of scalability to standard vision benchmarks or logic reasoning tasks. We propose the *Winfree Oscillatory Neural Network* (*WONN*), a dynamical neural architecture based on generalized Winfree dynamics. *WONN* evolves representations on the torus $(S^1)^d$ through structured oscillatory interactions, combining phase-based inductive biases with flexible and hierarchical interaction mechanisms instantiated as either fixed trigonometric mappings or learnable neural networks. We evaluate *WONN* on image recognition and complex reasoning tasks, including CIFAR, ImageNet, Maze-hard, and Sudoku. Across these domains, *WONN* achieves competitive or superior performance with strong parameter efficiency. **In particular, *WONN* is, to our knowledge, the first synchronization-based oscillatory architecture to scale competitively to ImageNet-1K.** Furthermore, on Maze-hard, *WONN* achieves **80.1% accuracy using only 1% of the parameters of prior state-of-the-art models**. These results suggest that structured oscillatory dynamics provide a scalable and parameter-efficient alternative to conventional neural architectures. The project page is available at `https://jiawen-dai.github.io/WONN_Project_Page/`.

1 Introduction

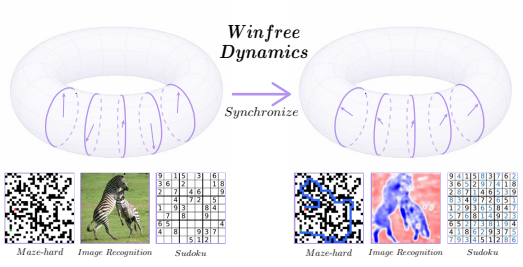


Figure 1: Illustration of the proposed *WONN*. The network evolves on the toroidal phase space $(S^1)^d$, where neurons are represented as phase oscillators. Through generalized Winfree synchronization dynamics, oscillators self-organize into structured collective states, enabling effective computation for both image recognition and reasoning tasks.

Oscillations and synchronization are ubiquitous mechanisms for organizing collective dynamics in biological and physical systems [2, 46, 20]. In neuroscience, oscillatory activity has long been associated with temporal coordination, inter-area communication, and structured information processing [2, 8, 27], while synchronization and temporal correlation have been proposed as mechanisms for perceptual binding and coherent representation formation [42, 43, 34]. In physics, coupled oscillator systems provide canonical examples of how simple local interactions can self-organize into coherent macroscopic behavior [46, 20]. Together, these observations suggest an alternative perspective on neural computation: rather than viewing representation learning purely as a sequence of static feature

[†] Correspondence to: yue-song@mail.tsinghua.edu.cn

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

transformations, computation may instead emerge from the collective evolution and synchronization of interacting dynamical states.

Despite their conceptual appeal, synchronization-based principles have so far had limited impact on mainstream machine learning. Existing synchronization-inspired approaches are primarily restricted to specialized settings such as object-centric representation learning or unsupervised object discovery [24, 39, 10, 23]. More recent architectures such as *AKOrN* [26] introduce Kuramoto-type oscillatory neurons into neural networks, but the empirical scalability of oscillatory architectures remains largely unclear. In particular, it is unknown whether synchronization-based computation can scale to large-scale image recognition benchmarks such as ImageNet-1k [4], or whether oscillatory dynamics can provide effective inductive biases for difficult combinatorial reasoning tasks such as maze pathfinding [21, 44]. This raises a fundamental question: *Can synchronization-driven dynamics serve as a scalable and parameter-efficient alternative to conventional feed-forward or transformer-style neural architectures?*

In this work, we introduce the *Winfree Oscillatory Neural Network* (*WONN*), a neural architecture built upon generalized Winfree synchronization dynamics [46, 25]. As illustrated in Fig. 1, *WONN* evolves neural representations on a high-dimensional toroidal phase space $(S^1)^d$, where each neuron is represented as a phase oscillator. Unlike Kuramoto-type models based primarily on pairwise phase differences, Winfree dynamics decompose interactions into separable sensitivity and influence functions, enabling more flexible synchronization behaviors and interaction structures.

WONN combines geometric inductive biases with flexible synchronization dynamics. The interaction functions can either be instantiated as fixed trigonometric mappings or parameterized by learnable neural networks, while grouped dynamics introduce hierarchical interactions that support both local coordination and global information flow. Under trigonometric interactions, the oscillatory core further admits an interaction energy that provides a useful stability perspective and diagnostic signal for reasoning tasks. Moreover, the dual phase–frequency state design separates fast synchronization dynamics from slower frequency evolution, enabling stable iterative refinement of representations.

We evaluate *WONN* on both visual perception and structured reasoning tasks. Across CIFAR and ImageNet benchmarks, *WONN* achieves competitive or superior accuracy compared with convolutional, transformer-based, and synchronization-inspired baselines while using substantially fewer parameters. **Importantly, *WONN* is, to our knowledge, the first synchronization-based oscillatory architecture to scale to ImageNet-1K with competitive performance against modern architectures such as ResNet and ViT.** The learned representations further exhibit characteristic bimodal phase organization, where distinct synchronized phase groups specialize to complementary coarse and fine image structures. On complex reasoning tasks, ***WONN* achieves competitive Maze-hard performance using only 1% of the parameters of the state-of-the-art *HRM* [44]**, while also reaching perfect test accuracy on Sudoku with strong parameter efficiency. Taken together, these results suggest that generalized Winfree synchronization dynamics provide a scalable, high-performance, and parameter-efficient computational principle for both visual perception and logical reasoning.

2 Winfree Oscillatory Neural Networks

This section develops *WONN* by connecting classical Winfree dynamics with learnable neural architectures. We first review the underlying Winfree model and its generalizations (Sec. 2.1), then introduce the *WONN* architecture with structured interactions (Sec. 2.2), and finally discuss its key dynamical and representational properties (Sec. 2.3).

2.1 From Winfree Dynamics to Learnable Oscillatory Systems

Winfree Dynamics. We consider a population of coupled phase oscillators that are governed by the Winfree model [46]:

$$\dot{\theta}_i = \omega_i + \frac{\kappa}{N} S(\theta_i) \sum_{j=1}^{N} I(\theta_j), \quad i = 1, \dots, N, \tag{1}$$

where $\theta_i \in \mathbb{T} := \mathbb{R}/(2\pi\mathbb{Z})$ is the phase of the i-th oscillator, $\omega_i \in \mathbb{R}$ is its natural frequency and $\kappa \in \mathbb{R}_+$ is the coupling strength. The interaction is mediated through a mean-field aggregation of the influence function $I(\cdot)$, modulated by the sensitivity function $S(\cdot)$.

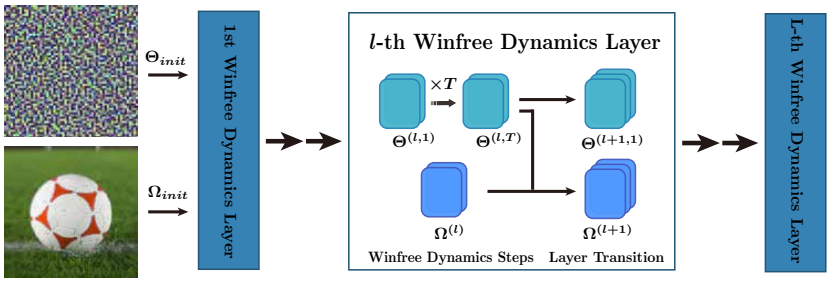


Figure 2: Overview of the *WONN* architecture. The input is encoded into an initial frequency state $\boldsymbol{\Omega_{init}}$, while the phase state $\boldsymbol{\Theta_{init}}$ is initialized randomly. Computation proceeds through stacked Winfree dynamics layers, each consisting of multiple parameter-shared recurrent dynamics steps followed by layer-transition updates of both phase and frequency states. Through this iterative synchronization process, *WONN* evolves structured oscillatory representations across layers on the toroidal phase space before producing the final prediction.

Structured Interactions Beyond Kuramoto. A widely studied special case of oscillator synchronization is the Kuramoto model, whose dynamics are governed by pairwise phase differences:

$$\dot{\theta}_i = \omega_i + \gamma \sum_{j=1}^{N} K_{ij} \sin(\theta_j - \theta_i), \quad t > 0, \tag{2}$$

where $\gamma \in \mathbb{R}_+$ denotes the coupling strength. These dynamics preserve global phase-shift symmetry under $\theta_i \mapsto \theta_i + \alpha$, since the interaction depends only on relative phase differences. While this symmetry simplifies analysis, it may also limit the diversity of attainable collective dynamics.

By contrast, the Winfree formulation allows more general interaction structures and naturally extends beyond this symmetric regime. In particular, introducing additional coupling terms yields

$$\dot{\theta}_i = \omega_i + \gamma \sum_{j=1}^{N} K_{ij} \big[\sin(\theta_j - \theta_i) + q \sin(\theta_j + \theta_i) \big], \tag{3}$$

where q controls the degree of symmetry breaking. This formulation recovers the Kuramoto model when $q = 0$, and produces asymmetric interactions for $q \neq 0$, enabling richer synchronization patterns. When $q = 1$, the dynamics can be rewritten as $\dot{\theta}_i = \omega_i + 2\gamma \cos\theta_i \sum_{j=1}^{N} K_{ij} \sin(\theta_j)$, which corresponds to a separable Winfree form with $S(\theta) = \cos(\theta)$ and $I(\theta) = \sin(\theta)$, arising from the first-order Fourier components of the interaction functions [25].

These formulations highlight a key perspective: *collective dynamics are determined by the structure of interactions rather than pairwise phase differences alone*. This flexibility makes Winfree-type systems a natural candidate for learnable neural architectures.

Geometric structure. Each oscillator evolves on the circle $\mathbb{T} := \mathbb{R}/(2\pi\mathbb{Z}) \cong S^1$, and the full system resides on the torus

$$\boldsymbol{\Theta} = (\theta_1, \ldots, \theta_d) \in \mathcal{M} := \mathbb{T}^d \cong (S^1)^d. \tag{4}$$

The dynamics define a vector field on $\mathcal{M}$, while updates can be locally represented in the tangent space $T_{\boldsymbol{\Theta}}\mathcal{M} \cong \mathbb{R}^d$. In practice, this suggests performing computations in a Euclidean embedding while enforcing periodic structure in the phase variables.

2.2 Winfree Oscillatory Neural Network

Fig. 2 illustrates the overall architecture of *WONN*, where representations are computed through iterative synchronization dynamics between coupled oscillatory neurons.

State and initialization. Each layer maintains a dual state consisting of a phase variable $\boldsymbol{\Theta} \in (S^1)^d$ and a frequency variable $\boldsymbol{\Omega} \in \mathbb{R}^d$. The phase captures oscillatory structure, while the frequency carries input-dependent information. Given an input x, we initialize $\boldsymbol{\Omega}^{(0)} = \boldsymbol{\Omega_{init}} = f_{\text{init}}(x)$ and $\boldsymbol{\Theta}^{(0)} = \boldsymbol{\Theta_{init}} \sim \mathcal{N}(0, \sigma^2)$, where f_{init} is a learnable embedding.

Discrete Winfree dynamics. At the core of *WONN* is a discretized Winfree evolution. For each layer l, we perform T recurrent updates:

$$\theta_i^{(l,t+1)} = \theta_i^{(l,t)} + \gamma\Big[\omega_i^{(l)} + S(\theta_i^{(l,t)}) \sum_j c_{ij} I(\theta_j^{(l,t)})\Big], \quad t = 1, \dots, T. \tag{5}$$

with parameters shared across time steps. Here c_{ij} denotes the coupling coefficient between oscillators i and j. The above ODE defines a controlled dynamical system over phases.

Interaction mechanisms. The behavior of the system is governed by the parameterization of the sensitivity and influence functions S and I. In the simplest case, these are defined point-wise as $S(\theta_i)$ and $I(\theta_i)$, implemented either using fixed trigonometric functions or learnable neural mappings.

To capture structured interactions, we introduce a ***grouped formulation***. As shown in Fig. 3, oscillators are partitioned into spatial patches $\mathcal{G}_{p,q}$, where each patch aggregates local states into a shared influence signal:

$$I_{p,q}^{\text{patch}} = I_{\text{patch}}(\{\theta_{i,j}\}_{(i,j)\in\mathcal{G}_{p,q}}). \tag{6}$$

Here $\mathcal{G}_{p,q}$ denotes an $N \times N$ region, where N is the group size. This induces a hierarchical interaction pattern, combining intra-group synchronization with inter-group coordination.

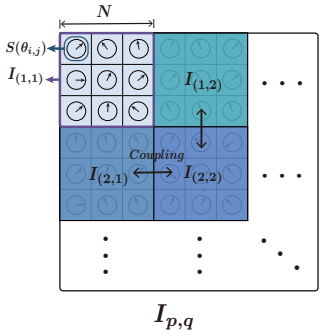


Figure 3: Illustration of S and I.

More generally, the coupling coefficients $\sum c_{ij}$ can be implemented through local or global components. Local interactions are implemented via convolutional operators with bounded receptive fields, while global interactions are realized through attention mechanisms. This results in a heterogeneous, state-dependent mean field, extending the homogeneous coupling in classical Winfree dynamics. We use global attentive coupling for all *WONN* models throughout the paper, and discuss local convolutional coupling as an alternative in Appendix C.2.

$$\mathcal{G}_{p,q} = \left\{(i,j) \in \mathbb{Z}^2 \;\middle|\; (p-1)N + 1 \le i \le pN,\ (q-1)N + 1 \le j \le qN\right\}. \tag{7}$$

Algorithm 1 Forward pass of *WONN*

Input: input x, layers L, steps $\{T_l\}_{l=1}^{L}$, step size γ, group size N
Output: model output y
$\Omega^{(1)} \leftarrow f_{\text{init}}(x)$, $\Theta^{(1,1)} \sim \mathcal{N}(0, \sigma^2 I)$
for $l = 1, \dots, L$ **do**
  for $t = 1, \dots, T_l$ **do**
    (1) Point-wise Sensitivity Function
    $S^{(l,t)} \leftarrow S_l(\Theta^{(l,t)})$
    (2) Patch-wise Influence Function
    partition $\Theta^{(l,t)}$ into non-overlapping $N \times N$ groups $\{\mathcal{G}_{p,q}\}$
    $I_{p,q}^{(l,t)} \leftarrow I_{\text{patch}}^{l}\Big(\{\theta_{i,j}^{(l,t)}\}_{(i,j)\in\mathcal{G}_{p,q}}\Big)$
    (3) Winfree Dynamics Step
    $M_{p,q}^{(l,t)} \leftarrow \sum_j c_{ij} I_j^{(l,t)}$
    $\widetilde{\mathbf{M}}^{(l,t)} \leftarrow \texttt{Broadcast}\Big(\mathbf{M}_{p,q}^{(l,t)}\Big)$ ▷ repeat within each group
    $\Delta\Theta^{(l,t)} \leftarrow \Omega^{(l)} + S^{(l,t)} \odot \widetilde{\mathbf{M}}^{(l,t)}$
    $\Theta^{(l,t+1)} \leftarrow \text{wrap}_{[-\pi,\pi)}\big(\Theta^{(l,t)} + \gamma\Delta\Theta^{(l,t)}\big)$
  end for
  (4) Layer Transition
  $\Theta^{(l+1,1)} \leftarrow \texttt{Theta-Update}_l(\Theta^{(l,T_l)})$, $\Omega^{(l+1)} \leftarrow \texttt{Omega-Update}_l(\Omega^{(l)}, \Theta^{(l,T_l)})$
end for
(5) Output
$y \leftarrow \texttt{OutputHead}(\Theta^{(L,T_L)})$
return y

Layer transitions. Within each layer, $\Omega^{(l)}$ remains fixed during the T phase updates in Eq. (5). After these updates, the phase and frequency are jointly updated as

$$\mathbf{\Theta}^{(l+1)} = \texttt{Theta-Update}(\mathbf{\Theta}^{(l,T)}), \quad \mathbf{\Omega}^{(l+1)} = \texttt{Omega-Update}(\mathbf{\Omega}^{(l)}, \mathbf{\Theta}^{(l,T)}). \tag{8}$$

This design separates fast phase evolution within each layer from slower updates of the frequency state across layers, allowing oscillatory patterns to be progressively refined.

Since phases lie on the circle, computations are performed via Euclidean embeddings while preserving periodic structure. Each phase is represented as $(\sin\theta, \cos\theta)$, and the phase update `Theta-Update` first applies weight-tied convolutional operators in this embedding space (which can be interpreted as linear operators acting on complex representation $e^{i\mathbf{\Theta}}$) and maps the result back to the torus via

$$\mathbf{\Theta}_{\text{new}} = \texttt{atan2}\big(\texttt{Conv}(\sin\mathbf{\Theta}), \texttt{Conv}(\cos\mathbf{\Theta})\big) \tag{9}$$

For frequency updates, we similarly construct a Euclidean embedding of Θ and combine it with the current frequency state:

$$\begin{aligned} \texttt{ThetaEmbedding}^{(l)}(\Theta) &= \texttt{Conv}_{\Theta}^{(l)}\left(\texttt{Concat}[\sin\Theta, \cos\Theta]\right), \\ \Omega^{(l+1)} &= \texttt{Conv}_{\Omega}^{(l)}\left(\texttt{Concat}\left[\texttt{ThetaEmbedding}^{(l)}(\Theta^{(l,T)}), \Omega^{(l)}\right]\right). \end{aligned} \tag{10}$$

Here, $\texttt{Conv}_{\Omega}^{(l)}$ denotes the learnable convolutional fusion block, including normalization and activation. The final model representation is obtained from the phase variables $\mathbf{\Theta}^{(L)}$ using a task-specific neural network head. Alg. 1 summarizes a single forward pass of our *WONN*.

2.3 Dynamical and Structural Properties of *WONN*

WONN exhibits several distinctive properties that emerge from its oscillatory dynamics and structured interaction design. We highlight the key properties below.

Oscillatory inductive bias. *WONN* imposes a geometric inductive bias by evolving representations on the torus $(S^1)^d$, enforcing periodic structure in the representation space. This encourages phase-based organization of features, where information can be encoded through relative phase configurations. Empirically, we observe that oscillators tend to form structured phase distributions, often exhibiting multi-modal patterns (*e.g.,* phase bifurcation with modes separated by π). Such phase organization may correlate with meaningful feature extraction. For instance in vision tasks, we observe that different phase modes align with distinct components such as object edges (see Sec. 3.2).

Hierarchical interaction structure. The grouped Winfree formulation introduces a two-level interaction hierarchy. Within each group, oscillators synchronize through shared local aggregation, while interactions across groups enable global coordination. *This structure interpolates between purely local and fully global coupling, controlled by the group size N.* In the limiting case $N{=}1$, the model reduces to point-wise interactions, whereas larger groups introduce shared contextual influence. This hierarchical design allows *WONN* to adapt its interaction scale to different tasks, balancing locality and global coherence.

Flexible interaction parameterization. *WONN* allows flexible choices of the sensitivity and influence functions S and I. In particular, these functions can be instantiated either as fixed trigonometric mappings (*e.g.,* $S(\theta){=}\cos\theta$, $I(\theta){=}\sin\theta$) or as learnable neural networks (*e.g.,* MLPs). The trigonometric parameterization provides a structured and stable interaction form aligned with classical oscillator models, while the parameterization of learnable neural networks increases expressivity. This flexibility allows to trade off inductive bias and representational capacity across different tasks.

Energy structure and stability. The phase dynamics of *WONN* inherit stability properties from the underlying Winfree system. Due to the periodic geometry of $(S^1)^d$, the natural frequency term cannot, in general, be associated with a globally defined potential (see Appendix D). However, when the natural frequency is excluded, the coupling is fixed and symmetric, and the interaction functions take separable trigonometric forms, the dynamics admit a Lyapunov function that decreases along trajectories, leading to stable synchronization patterns and attractors in phase space. In this regime, the interaction energy can be written as $E_{\text{int}} = -\sum_i \sin\theta_i \sum_j c_{ij}\sin(\theta_j)$, capturing phase alignment under the coupling coefficients. Although this continuous-time analysis does not provide a general convergence guarantee for the full learned *WONN*, the interaction energy serves as a practical

diagnostic: in Maze reasoning tasks (Sec. 3.3), we use it as a proxy for solution quality and perform voting to improve performance.

Separation of timescales. *WONN* separates fast oscillatory dynamics from slower feature evolution through its dual-state design. Within each layer, the phase state $\mathbf{\Theta}$ undergoes multiple recurrent updates, enabling rapid synchronization and local reorganization on $(S^1)^d$. In contrast, the frequency state $\mathbf{\Omega}$ is updated once per layer, acting as a slower carrier of information across layers. This separation of timescales decouples transient oscillatory computation from cross-layer feature transformation, which is reminiscent of hierarchical processing in biological systems [13, 15, 9, 44].

3 Experiments

3.1 Experimental Setup

We evaluate *WONN* on a diverse set of benchmarks spanning visual perception and logical reasoning. The visual tasks include CIFAR-10/100 [19] and ImageNet-100/1K [4], while the reasoning tasks consist of Maze-hard pathfinding [21, 44] and Sudoku solving [45].

Table 1: Image classification results on **CIFAR-10/100**. † ViT models are trained with additional regularization (e.g., CutMix and label smoothing), while other models follow a plain training protocol.

Models	CIFAR-10		CIFAR-100	
	Accuracy (%)	**# Parameters**	**Accuracy (%)**	**# Parameters**
ResNet-18	93.48 ± 0.16	11.17M	70.53 ± 0.10	11.22M
ResNet-50	94.22 ± 0.14	23.52M	73.54 ± 0.41	23.71M
ViT-T†	90.34 ± 0.18	5.36M	67.59 ± 0.44	5.38M
ViT-S†	93.43 ± 0.29	21.34M	71.18 ± 0.62	21.38M
ViT-B†	92.04 ± 0.25	85.15M	71.05 ± 0.40	85.22M
AKOrN^{attn}	93.66 $\pm$ 0.17	4.60M	72.03 $\pm$ 0.34	4.62M
$S(\theta_i)\&I(\theta_i)$ *as MLPs*				
WONN (Ch $= 128 \to 128$)	94.55 $\pm$ 0.09	3.08M	73.77 $\pm$ 0.40	3.09M
WONN (Ch $= 64 \to 256$)	95.12 $\pm$ 0.04	7.54M	75.12 $\pm$ 0.49	7.56M
WONN (Ch $= 256 \to 256$)	**95.24** $\pm$ 0.12	12.02M	**76.20** $\pm$ 0.45	12.04M
$S(\theta_i)\&I(\theta_i)$ *as trigonometric functions*				
WONN (Ch $= 128 \to 128$)	94.50 ± 0.15	2.98M	74.48 ± 0.16	3.00M
WONN (Ch $= 64 \to 256$)	95.08 ± 0.07	7.40M	75.81 ± 0.33	7.43M
WONN (Ch $= 256 \to 256$)	**95.26** ± 0.05	11.84M	**76.17** ± 0.52	11.86M

Visual Perception. For image classification, we benchmark *WONN* against representative convolutional and transformer-based architectures, including *ResNet-18* and *ResNet-50* [14], as well as Vision Transformers (*ViT-Tiny/Small/Base*) [5, 41]. We also include *AKOrN* [26], a recent synchrony-based model, as a closely related baseline. ***All methods are trained under standard, architecture-appropriate protocols, and no additional training techniques are introduced to favor WONN, ensuring a fair comparison.***

Logical Reasoning. For reasoning tasks, we evaluate *WONN* on Maze-hard and Sudoku. On Maze-hard, we mainly compare against both general-purpose LLMs (*e.g., DeepSeek-R1* [11]) and specialized recurrent reasoning architectures, including *HRM* [44] and *TRM* [16]. On Sudoku, we compare with *AKOrN* [26] and several other neural reasoning baselines [45, 30, 48].

Other implementation details and extended experimental results are referred to in Appendix C.

3.2 Image Classification on CIFAR and ImageNet

We evaluate *WONN* on standard image classification benchmarks, including CIFAR-10/100 and ImageNet-100/1K. We consider two parameterizations of the interaction functions S and I: (i) learnable MLP-based mappings and (ii) fixed trigonometric mappings. All models use grouped Winfree dynamics in Eq. (6) with group size $N = 2$. To assess the scalability of model size, we study three model sizes with approximately 3M, 7.5M, and 12M parameters.

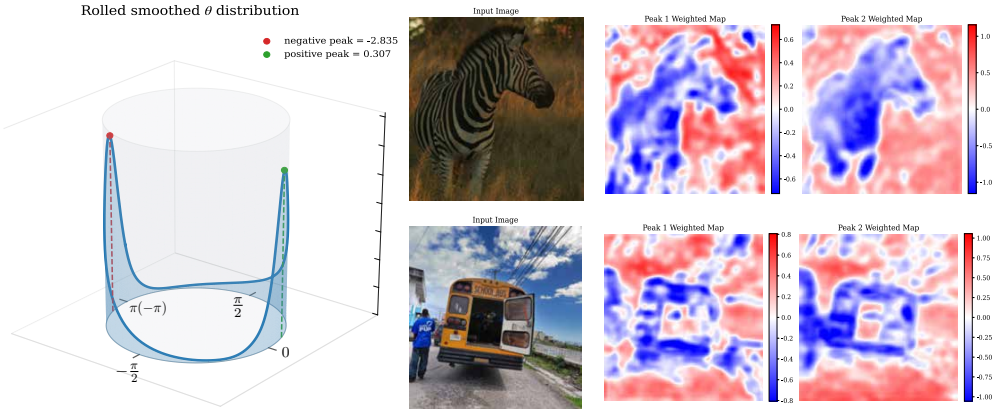


Figure 4: Visualization of the final phase distribution in *WONN* on image recognition tasks. **Left:** the phase variables θ exhibit a characteristic bimodal distribution, with two dominant peaks approximately separated by π. **Right:** heatmaps constructed by weighting oscillators according to their proximity to each phase peak. The two modes capture complementary object structures: one emphasizes coarse global regions, while the other focuses on finer local details and boundaries. Together, the synchronized phase organization produces structured object-aware representations.

Results on CIFAR-10/100. Table 1 compares *WONN* with *ResNet*, *ViT*, and *AKOrN*, with results averaged over three random seeds. *WONN* consistently outperforms all baselines on both CIFAR-10 and CIFAR-100 while using substantially fewer parameters. Notably, *WONN*, trained with a plain protocol, surpasses transformer-based baselines that rely on additional regularization. These results demonstrate the effectiveness and parameter efficiency of oscillatory dynamics.

Results on ImageNet-100/1K. Table 2 shows that the trends observed on CIFAR persist at larger scale: *WONN* achieves competitive accuracy with substantially fewer parameters than standard baselines. In particular, *WONN* achieves comparable ImageNet-1K accuracy to *ResNet-50* while using approximately half the parameters (12.28M vs. 25.56M), and outperforms the evaluated *ViT* baselines. ***To the best of our knowledge, this is the first synchrony-based architecture that scales effectively to large-scale image classification.***

Table 2: Image classification results on **ImageNet-100** and **ImageNet-1K**. †ViT models are trained with additional regularization techniques compared to other models. Underlining denotes our reimplementation.

Models	**ImageNet-100**		**ImageNet-1k**	
	Accuracy(%)	**# Parameters**	**Accuracy(%)**	**# Parameters**
ResNet-18	78.20	11.23M	69.73	11.69M
ResNet-50	81.18	23.71M	**76.89**	25.56M
ViT-S-16†	77.96	21.70M	75.54	22.05M
ViT-B-16†	76.36	85.87M	75.85	86.57M
AKOrN^{attn}	<u>80.08</u>	4.62M	<u>67.45</u>	4.85M
$S(\theta_i)\&I(\theta_i)$ *as MLPs*				
WONN (Ch $= 128 \rightarrow 128$)	81.50	3.09M	–	–
WONN (Ch $= 64 \rightarrow 256$)	82.04	7.56M	74.84	7.79M
WONN (Ch $= 256 \rightarrow 256$)	**82.88**	12.05M	**76.80**± 0.14	12.28M
$S(\theta_i)\&I(\theta_i)$ *as trigonometric functions*				
WONN (Ch $= 128 \rightarrow 128$)	81.76	3.00M	–	–
WONN (Ch $= 64 \rightarrow 256$)	**82.56**	7.43M	–	–
WONN (Ch $= 256 \rightarrow 256$)	82.22	11.87M	76.36	12.10M

Two-peak phase distributions. To better understand the internal phase organization of *WONN*, we analyze the final θ distribution in Fig. 4 left. Interestingly, the phase variables consistently bifurcate into two dominant modes, producing a characteristic bimodal distribution with peaks approximately

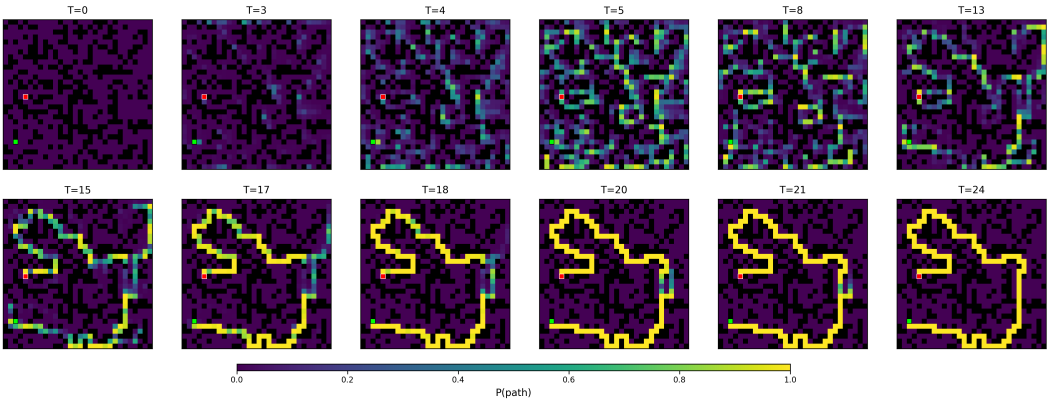


Figure 5: Synchronous path formation on Maze-hard. Each panel shows the predicted per-cell path probability. *WONN* initially produces multiple diffuse path fragments. As the dynamics evolve, these fragments progressively synchronize and merge, forming a coherent path from start to goal.

separated by π. To visualize the functional role of these modes, we construct weighted activation maps by measuring the proximity of each oscillator phase to the corresponding peak.

Fig. 4 right reveals that the two synchronized phase modes capture *complementary aspects of the object structure.* One phase mode primarily responds to coarse, global object regions, while the other emphasizes finer local structures and boundaries. Despite different spatial characteristics, both modes remain highly aligned with semantically relevant object regions. This suggests that the oscillatory synchronization dynamics naturally induce a structured phase decomposition of representations, where different synchronized phase groups specialize to distinct levels of spatial abstraction.

Notably, the emergence of such bimodal phase organization is not explicitly imposed by any architecture or supervision objective, but instead arises spontaneously from the collective synchronization dynamics. We observe this bimodal organization across CIFAR-10, CIFAR-100, ImageNet-100, and ImageNet-1K, and hypothesize that this self-organized phase separation contributes to the strong recognition performance and parameter efficiency of *WONN* by enabling coordinated yet diverse feature representations across oscillators. The mechanism underlying its emergence and whether it is preferable to other multimodal organizations remain open questions. Additional visualizations are provided in Appendix E.1.

3.3 Maze-hard Pathfinding

We evaluate the reasoning capability of *WONN* on the **Maze-hard** benchmark, which requires finding optimal paths in 30×30 mazes and tests long-horizon search and algorithmic reasoning. Following [44], we construct $1,000$ training and $1,000$ test instances whose difficulty levels exceed 110. For this task, *WONN* uses point-wise interactions ($N = 1$).

Results on Maze-hard. Table 3 shows that *WONN* significantly outperforms LLM-based baselines and the synchrony-based model *AKOrN*, and surpasses the strong recurrent model *HRM*. Despite using only **0.396M** parameters (**1.47%** of *HRM*), *WONN* achieves **76.2%** accuracy. Importantly, leveraging the energy structure of the dynamics, we further improve performance via *energy voting*: by sampling 32 trajectories and selecting the solution with the lowest final energy, accuracy increases to **80.1%**, compared with 85.3% for the 6.823M-parameter TRM-Att. This improvement highlights the practical usage of the energy formulation as a *test-time selection criterion*, while the comparison with TRM-Att demonstrates a favorable accuracy–parameter-efficiency trade-off. Taken together, these results demonstrate that structured oscillatory dynamics provide an effective and highly parameter-efficient inductive bias for long-horizon reasoning.

Synchronous Pathfinding on Maze-hard. To better understand the role of oscillatory dynamics, we visualize the intermediate states of *WONN* on Maze-hard (Fig. 5). We plot heatmaps of the predicted per-cell path probability, revealing how solutions emerge over time.

Table 3: Maze-hard pathfinding results. *Energy Voting* denotes selecting the solution with the lowest energy among 32 independently sampled trajectories.

Models	Accuracy (%)	# Parameters
LLMs		
Deepseek R1	0.0	671B
Claude 3.7 8K	0.0	?
O3-mini-high	0.0	?
Recurrent models		
HRM	74.5	27M
TRM-Att (dihedral augmented)	**85.3**	7M
TRM-MLP (dihedral augmented)	0.0	19M
Other synchrony-based model		
AKOrN^{attn}	36.2	1M
$S(\theta_i)$ & $I(\theta_i)$ *as trigonometric functions*		
WONN (K=1)	**76.2**	**0.396M**
WONN (Energy Voting, K=32)	**80.1**	**0.396M**

Unlike conventional methods that construct paths incrementally or greedily, *WONN* exhibits a distinct synchronization-driven process. In the early stage, the model generates multiple diffuse path fragments across the maze that correspond to candidate solution trajectories. These fragments are not locally consistent and may include invalid transitions. As the dynamics evolve, these candidate fragments begin to interact and synchronize. Inconsistent paths are suppressed, while compatible segments reinforce each other and progressively merge. This collective refinement process leads to the emergence of a single coherent path connecting the start and goal. In later stages, the model further refines the solution by removing redundant segments and improving path optimality.

Additional examples (Appendix E.2) show that this multi-fragment-to-single-path behavior is consistent across hard instances. These observations suggest that synchronization dynamics provide a powerful inductive bias for combinatorial search: *instead of committing to a single trajectory early, the model explores multiple candidates in parallel and resolves them through collective alignment.*

3.4 Sudoku Reasoning

We further evaluate *WONN* on Sudoku, a standard benchmark for combinatorial reasoning. We use the dataset from [45], where training instances contain 31–42 given digits, and evaluation is performed on 1,000 test boards drawn from the same distribution. For this task, *WONN* adopts point-wise interactions ($N = 1$) and follows the same training protocol as *AKOrN*.

Results on Sudoku. Table 4 reports results averaged over five runs. *WONN* achieves perfect accuracy on the test set, matching the best-performing methods. It improves upon *AKOrN* and *HRM* while using significantly fewer parameters, and achieves comparable performance to both recurrent and transformer-based baselines [30, 48]. Consistent with the Maze-hard results, these findings suggest that oscillatory dynamics provide an effective inductive bias for structured reasoning tasks.

Table 4: Sudoku solving results.

Models	Accuracy (%)	# Parameters
SAT-Net	98.3	0.618M
RRN	99.8	0.201M
R-Transformer	**100.0**	0.211M
Transformer	98.6$\pm$ 0.3	16.87M
HRM	99.7$\pm$ 0.2	27.28M
AKOrN$^{\text{attn}}(T = 16)$	99.8$\pm$ 0.1	2.98M
$S(\theta_i)\&I(\theta_i)$ *as trigonometric functions*		
WONN $(T = 16)$	**100.0**$\pm$ **0.0**	1.58M

3.5 Computational Cost

To evaluate efficiency beyond parameter count, we benchmark training latency, single-sample inference latency, and peak GPU memory on CIFAR-10, Sudoku, and Maze-hard. All measurements are conducted on a single NVIDIA H100 using the same numerical precision and timing protocol. Training latency is measured with batch size 32.

Table 5 compares *WONN* with representative feed-forward and oscillatory baselines on CIFAR-10. As expected for a dynamical architecture, *WONN* incurs additional latency relative to ResNet-50 and ViT-S/4 while using roughly half their parameters. Although *AKOrN* uses fewer parameters, *WONN* is faster and more memory-efficient during training. Additional measurements on Sudoku and Maze-hard, including the cost of energy voting, are provided in Appendix C.5.

Table 5: Computational costs on CIFAR-10, measured on a single NVIDIA H100 with batch sizes of 32 for training and 1 for inference. *WONN* uses trigonometric interactions with 256 channels.

Model	**Params.**	**Latency (ms)**		**Peak memory (GiB)**	
		Train/batch	Infer/sample	Train	Infer
ResNet-50	23.52M	44.37	1.71	0.765	0.125
ViT-S/4	21.34M	37.52	1.87	0.690	0.113
AKOrN	4.60M	59.61	3.60	5.243	0.083
WONN	11.84M	51.74	3.67	3.445	0.084

4 Conclusion

We introduced the *Winfree Oscillatory Neural Network* (*WONN*), a neural architecture built upon generalized Winfree synchronization dynamics. Unlike conventional architectures that primarily rely on static feature transformations, *WONN* performs computation through the collective evolution and synchronization of phase oscillators on a toroidal state space $(S^1)^d$. By combining flexible interaction parameterizations, hierarchical grouped synchronization dynamics, and a dual phase–frequency state design, *WONN* provides a scalable framework for oscillatory neural computation.

Empirically, we show that synchronization-based architectures can scale far beyond the limited settings previously explored in oscillatory or synchronization-inspired models. On visual perception tasks, *WONN* achieves competitive or superior performance on CIFAR and ImageNet benchmarks while maintaining strong parameter efficiency. In particular, *WONN* successfully scales to ImageNet-1K, which, to the best of our knowledge, makes it the first synchronization-based oscillatory architecture to achieve competitive large-scale image recognition performance. The learned phase representations further exhibit characteristic bimodal phase organization, where distinct synchronized phase groups specialize to complementary coarse and fine image structures. Beyond perception, *WONN* also demonstrates strong capability on structured reasoning tasks. On Maze-hard, *WONN* achieves competitive performance using only a tiny fraction of the parameters of state-of-the-art recurrent reasoning models. Visualization of the temporal evolution further reveals that the network progressively assembles local candidate path fragments into coherent global solutions, suggesting that synchronization provides an effective inductive bias for complex search and reasoning problems. On Sudoku, *WONN* further achieves perfect test accuracy with strong parameter efficiency.

More broadly, our results suggest that synchronization dynamics may provide a promising alternative computational paradigm for modern machine learning. We hope this work encourages further exploration of oscillatory neural computation, synchronization-driven representation learning, and dynamical systems perspectives on large-scale neural architectures.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (Grant No. 62676211). Yue Song thanks Max Welling for the fruitful discussion and his insightful feedback.

References

[1] Michael Breakspear, Stewart Heitmann, and Andreas Daffertshofer. Generative models of cortical oscillations: Neurobiological implications of the kuramoto model. *Frontiers in Human Neuroscience*, 4:190, 2010.

[2] György Buzsáki. *Rhythms of the Brain*. Oxford University Press, New York, 2006.

[3] Luke Darlow, Ciaran Regan, Sebastian Risi, Jeffrey Seely, and Llion Jones. Continuous thought machines. *arXiv preprint arXiv:2505.05522*, 2025.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[6] Katharina Duecker, Marco Idiart, Marcel van Gerven, and Ole Jensen. Oscillations in an artificial neural network convert competing inputs into a temporal code. *PLOS Computational Biology*, 20(9):e1012429, 2024.

[7] Felix Effenberger, Pedro Carvalho, Igor Dubinin, and Wolf Singer. The functional role of oscillatory dynamics in neocortical circuits: A computational perspective. *Proceedings of the National Academy of Sciences*, 122(4):e2412830122, 2025.

[8] Pascal Fries. A mechanism for cognitive dynamics: Neuronal communication through neuronal coherence. *Trends in Cognitive Sciences*, 9(10):474–480, 2005.

[9] Mehrshad Golesorkhi, Javier Gomez-Pilar, Shankar Tumati, Maia Fraser, and Georg Northoff. Temporal hierarchy of intrinsic neural timescales converges with spatial core-periphery organization. *Communications Biology*, 4(1):277, 2021.

[10] Anand Gopalakrishnan, Aleksandar Stanić, Jürgen Schmidhuber, and Michael Curtis Mozer. Recurrent complex-weighted autoencoders for unsupervised object discovery. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

[11] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645:633–638, 2025.

[12] Seung-Yeal Ha, Dongnam Ko, Jinyeong Park, and Sang Woo Ryoo. Emergent dynamics of winfree oscillators on locally coupled networks. *Journal of Differential Equations*, 260(5):4203–4236, 2016.

[13] Uri Hasson, Eunice Yang, Ignacio Vallines, David J. Heeger, and Nava Rubin. A hierarchy of temporal receptive windows in human cortex. *Journal of Neuroscience*, 28(10):2539–2550, 2008.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[15] Christopher J. Honey, Thomas Thesen, Tobias H. Donner, Lauren J. Silbert, Chad E. Carlson, Orrin Devinsky, Werner K. Doyle, Nava Rubin, David J. Heeger, and Uri Hasson. Slow cortical dynamics and the accumulation of information over long timescales. *Neuron*, 76(2):423–434, 2012.

[16] Alexia Jolicoeur-Martineau. Less is more: Recursive reasoning with tiny networks. *arXiv preprint arXiv:2510.04871*, 2025.

[17] T. Anderson Keller, Lyle Muller, Terrence J. Sejnowski, and Max Welling. Traveling waves encode the recent past and enhance sequence learning. In *International Conference on Learning Representations*, 2024.

[18] T. Anderson Keller and Max Welling. Neural wave machines: Learning spatiotemporally structured representations with locally coupled oscillatory recurrent neural networks. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 16168–16189. PMLR, 2023.

[19] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[20] Yoshiki Kuramoto. *Chemical Oscillations, Waves, and Turbulence*, volume 19 of *Springer Series in Synergetics*. Springer, Berlin, Heidelberg, 1984.

[21] Lucas Lehnert, Sainbayar Sukhbaatar, DiJia Su, Qinqing Zheng, Paul McVay, Michael Rabbat, and Yuandong Tian. Beyond a*: Better planning with transformers via search dynamics bootstrapping. In *First Conference on Language Modeling*, 2024.

[22] Jingkun Liu, Yisong Yue, Max Welling, and Yue Song. Krause synchronization transformers. *arXiv preprint arXiv:2602.11534*, 2026.

[23] Sindy Löwe, Phillip Lippe, Francesco Locatello, and Max Welling. Rotating features for object discovery. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[24] Sindy Löwe, Phillip Lippe, Maja Rudolph, and Max Welling. Complex-valued autoencoders for object discovery. *Transactions on Machine Learning Research*, 2022.

[25] M. Manoranjani, Shamik Gupta, D. V. Senthilkumar, and V. K. Chandrasekar. Generalization of the kuramoto model to the winfree model by a symmetry breaking coupling. *The European Physical Journal Plus*, 138(2):144, 2023.

[26] Takeru Miyato, Sindy Löwe, Andreas Geiger, and Max Welling. Artificial kuramoto oscillatory neurons. In *International Conference on Learning Representations*, 2025.

[27] Lyle Muller, Frédéric Chavane, John Reynolds, and Terrence J. Sejnowski. Cortical travelling waves: Mechanisms and computational principles. *Nature Reviews Neuroscience*, 19(5):255–268, 2018.

[28] Daniel Neil, Michael Pfeiffer, and Shih-Chii Liu. Phased LSTM: Accelerating recurrent network training for long or event-based sequences. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[29] Tuan Nguyen, Hirotada Honda, Takashi Sano, Vinh Nguyen, Shugo Nakamura, and Tan M. Nguyen. From coupled oscillators to graph neural networks: Reducing over-smoothing via a kuramoto model-based approach. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 2710–2718. PMLR, 2024.

[30] Rasmus Berg Palm, Ulrich Paquet, and Ole Winther. Recurrent relational networks. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[31] Matthew Ricci, Minju Jung, Yuwei Zhang, Mathieu Chalvidal, Aneri Soni, and Thomas Serre. Kuranet: Systems of coupled oscillators that learn to synchronize. *arXiv preprint arXiv:2105.02838*, 2021.

[32] T. Konstantin Rusch, Benjamin P. Chamberlain, James Rowbottom, Siddhartha Mishra, and Michael M. Bronstein. Graph-coupled oscillator networks. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 18888–18909. PMLR, 2022.

[33] T. Konstantin Rusch and Siddhartha Mishra. Coupled oscillatory recurrent neural network (coRNN): An accurate and (gradient) stable architecture for learning long time dependencies. In *International Conference on Learning Representations*, 2021.

[34] Wolf Singer and Charles M. Gray. Visual feature integration and the temporal correlation hypothesis. *Annual Review of Neuroscience*, 18:555–586, 1995.

[35] Yue Song, Andy Keller, Nicu Sebe, and Max Welling. Flow factorized representation learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[36] Yue Song, Andy Keller, Nicu Sebe, and Max Welling. Latent traversals in generative models as potential flows. In *International Conference on Machine Learning*. PMLR, 2023.

[37] Yue Song, T. Anderson Keller, Sevan Brodjian, Takeru Miyato, Yisong Yue, Pietro Perona, and Max Welling. Kuramoto orientation diffusion models. In *Advances in Neural Information Processing Systems*, 2025.

[38] Yue Song, Thomas Anderson Keller, Nicu Sebe, and Max Welling. *Structured representation learning: from homomorphisms and disentanglement to equivariance and topography*. Springer Nature, 2025.

[39] Aleksandar Stanić, Anand Gopalakrishnan, Kazuki Irie, and Jürgen Schmidhuber. Contrastive training of complex-valued autoencoders for object discovery. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

[40] Yuchen Tian, Samuel Tensingh, Jason K. Eshraghian, Nhan Duy Truong, and Omid Kavehei. Synchrony-gated plasticity with dopamine modulation for spiking neural networks. *Transactions on Machine Learning Research*, 2025.

[41] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers and distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357, 2021.

[42] Anne M. Treisman and Garry Gelade. A feature-integration theory of attention. *Cognitive Psychology*, 12(1):97–136, 1980.

[43] Christoph von der Malsburg. The correlation theory of brain function. Technical Report Internal Report 81-2, Max-Planck-Institute for Biophysical Chemistry, 1981.

[44] Guan Wang, Jin Li, Yuhao Sun, Xing Chen, Changling Liu, Yue Wu, Meng Lu, Sen Song, and Yasin Abbasi Yadkori. Hierarchical reasoning model. *arXiv preprint arXiv:2506.21734*, 2025.

[45] Po-Wei Wang, Priya Donti, Bryan Wilder, and Zico Kolter. SATNet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6545–6554, 2019.

[46] A. T. Winfree. Biological rhythms and the behavior of populations of coupled oscillators. *Journal of Theoretical Biology*, 16(1):15–42, 1967.

[47] Mingqing Xiao, Yansen Wang, Dongqi Han, Caihua Shan, and Dongsheng Li. Kuramoto oscillatory phase encoding: Neuro-inspired synchronization for improved learning efficiency. *arXiv preprint arXiv:2604.07904*, 2026.

[48] Zhun Yang, Adam Ishay, and Joohyung Lee. Learning to solve constraint satisfaction problems with recurrent transformer. In *International Conference on Learning Representations*, 2023.

Appendix Contents

A Related Work **15**

B Discussions **15**

B.1 Limitation and Future Work 15

B.2 Difference against *AKOrN* 15

C Additional Implementation Details and Extended Experiment Results **16**

C.1 Implementation Details 16

C.1.1 Image Recognition 16

C.1.2 Solving Maze-hard 18

C.1.3 Solving Sudoku 18

C.2 Additional Results and Ablations on Image Recognition 19

C.3 Fixed and Learnable Components 21

C.4 Solving Maze-hard 21

C.5 Computational Cost and Memory Usage 21

D Energy Structure and Topological Obstruction **22**

E Additional Visualization **24**

E.1 Additional Visualization on Image Recognition 24

E.2 Additional Visualization on Maze-hard 26

E.3 Additional Visualization on Sudoku 27

A Related Work

Oscillatory neural computation. Oscillatory activity is widely regarded as a fundamental organizing principle in biological neural systems, supporting temporal coordination, communication, and structured information processing [2, 8, 27]. This perspective has motivated the use of oscillatory dynamics as inductive biases in artificial neural networks, where oscillations provide a temporal scaffold for computation. Early work introduced rhythmic gating mechanisms for sparse and event-based processing [28], while more recent approaches have explored dynamical systems-based architectures, including stable recurrent networks and graph-based models, where oscillatory or wave-like dynamics enable long-range interactions and memory [33, 32, 18, 36, 35, 17]. More broadly, oscillatory mechanisms have been shown to improve learning dynamics, robustness, and representational organization by structuring computation through phase and temporal evolution [6, 7, 38].

Synchronization-inspired models in machine learning. While oscillations structure computation in time, synchronization governs how multiple oscillators interact through collective phase alignment. Building on this principle, a growing body of work has incorporated synchronization into neural architectures. Prior approaches have leveraged phase or complex-valued representations to organize object-centric features and latent structure [24, 39, 23, 10], introduced synchronization mechanisms into modern architectures through bounded-confidence interactions [22], and explored neural synchronization as a latent representation in recurrent dynamical architectures such as the Continuous Thought Machine [3]. Synchrony-based signals have also been used as training-time plasticity mechanisms in spiking neural networks, for example through dopamine-modulated spike-synchrony-dependent plasticity [40]. Closely related to our work are oscillator-based neural systems such as *AKOrN* [26]. At a more fundamental level, these approaches are closely related to classical coupled-oscillator systems, particularly the Kuramoto and Winfree models [20, 46]. Kuramoto-type dynamics based on pairwise phase interactions have been widely explored in cortical oscillation modeling, learnable oscillator systems, graph neural networks, orientation diffusion models, and phase-based neural architectures [1, 31, 29, 37, 47]. In contrast, Winfree-type formulations introduce separable sensitivity–influence interactions, enabling more flexible and expressive collective dynamics [12, 25]. *WONN* builds on this formulation by using generalized Winfree dynamics with flexible parameterizations and grouped interactions, enabling hierarchical information flow through synchronization dynamics on a toroidal phase space.

B Discussions

B.1 Limitation and Future Work

Our results suggest that oscillatory synchronization can serve not only as a biological or physical metaphor, but also as a practical computational principle for modern neural networks. At the same time, *WONN* represents only an initial step toward synchronization-driven large-scale neural computation. While the recurrent Winfree dynamics provide strong parameter efficiency and expressive iterative refinement, they also introduce additional computational cost compared with conventional feed-forward architectures such as ResNets and ViTs, since each layer involves multiple recurrent synchronization steps. Furthermore, although *WONN* demonstrates competitive scalability up to ImageNet-1K, exploring larger-scale models and even language models remains an important direction for future work. Future research may further investigate more efficient synchronization mechanisms, adaptive or continuous-time dynamics, larger-scale oscillatory architectures, and deeper theoretical understanding of the stability, geometry, and representation structure induced by Winfree-type neural dynamics.

B.2 Difference against *AKOrN*

WONN is closely related to *AKOrN* [26] in that both methods introduce oscillatory dynamics into neural architectures. However, the two models differ in several essential aspects.

Practical scalability. In Table 8, we additionally compare the practical training cost of *WONN* and *AKOrN* on ImageNet-1K. Despite using 8 poweful H200 GPUs, *AKOrN* requires nearly 1,000 GPU hours to train its standard-size model for only 200 epochs, whereas *WONN* is trained for 300 epochs using only 4 H100 GPUs with substantially lower total GPU hours. We also observe a large memory gap under the same per-GPU batch size: *AKOrN* requires more than 110GB aggregate GPU memory,

compared with approximately 35GB for *WONN*. These results suggest that *WONN* is considerably more favorable for large-scale training under realistic compute and memory constraints.

Different state geometries. *AKOrN* represents each oscillatory neuron by a high-dimensional unit vector, so its oscillator state is embedded in a Euclidean ambient space. In contrast, *WONN* represents the oscillator state directly by a phase angle. Thus, each scalar oscillator evolves on the circle S^1, and a collection of d oscillators naturally lives on the torus $(S^1)^d$. This makes the periodic geometry of the representation explicit rather than treating oscillation through a vector-valued Euclidean embedding.

Separable interaction mechanisms. *AKOrN* is motivated by Kuramoto-type synchronization (Eq. 2), where interactions are governed primarily by pairwise phase differences. Such dynamics are invariant under global phase shifts, since replacing every phase by $\theta_i + \alpha$ leaves the differences $\theta_j - \theta_i$ unchanged. While this symmetry simplifies the interaction structure, it may also restrict the class of admissible dynamics. In contrast, *WONN* builds upon the more general Winfree formulation (Eq. 1), where oscillator updates are decomposed into a sensitivity function of the receiving oscillator and an influence function of neighboring oscillators. This formulation recovers Kuramoto-style phase-difference coupling as a special case, while also permitting symmetry-breaking interactions (Eq. 3) that depend on absolute phase. As a result, *WONN* defines a more flexible and expressive class of oscillatory interactions.

Grouped hierarchical interactions. *WONN* extends the classical Winfree formulation into a grouped neural architecture. Instead of applying only pointwise oscillator interactions, *WONN* partitions oscillators into groups and computes group-level influence signals. This grouped formulation induces a hierarchical interaction structure: oscillators may synchronize locally within a group, while group-level influence signals coordinate information across larger spatial or feature regions. The group size therefore controls the interaction scale, interpolating between pointwise dynamics and more structured collective dynamics.

Flexible interaction parameterization. *WONN* provides a flexible parameterization of the Winfree interaction functions. The sensitivity and influence functions can either be instantiated as fixed trigonometric mappings, such as $S(\theta) = \cos\theta$ and $I(\theta) = \sin\theta$, or implemented as learnable neural networks. The former preserves a strong oscillator-inspired inductive bias, while the latter increases representational flexibility and expressiveness. This allows the interaction dynamics of *WONN* to adapt naturally to different tasks, datasets, and computational regimes.

C Additional Implementation Details and Extended Experiment Results

C.1 Implementation Details

C.1.1 Image Recognition

For Image Recognition, we train all the models under comparable training protocols. The detailed training recipes are reported in Table 6 for **CIFAR-10/100**, Table 7 for **ImageNet-100**, and Table 8 for **ImageNet-1K**.

Table 6: Implementation details on **CIFAR-10/100**.

Hyperparameters	**Models**				
	WONN	***AKOrN***^{attn}	**ResNet-18/50**	**ViT-T/S**	**ViT-B**
Batch Size	64	128	128	256	256
Optimizer	AdamW	Adam	AdamW	AdamW	AdamW
Weight Decay	0.0001	–	0.0001	0.05	0.05
LR	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	$1 \cdot 10^{-4}$
LR Decay	✗	✗	✗	cosine	cosine
Min LR	✗	✗	✗	0	0
Warmup Epochs	✗	✗	✗	10	10
Epochs	200	200	200	300	300
Dynamics Layers (L)	6	3	–	–	–
Dynamics Time Step (T)	3	3	–	–	–
Group Size (N)	2	–	–	–	–
Oscillator Dimension (n)	–	4	–	–	–
GPUs	1×H100	1×H100	1×H100	1×H100	1×H100
Dropout	✗	✗	✗	✗	✗
Label Smoothing	✗	✗	✗	0.1	0.1
Mixup Alpha	✗	✗	✗	0.8	0.8
Cutmix Alpha	✗	✗	✗	1.0	1.0
Mixup Prob.	✗	✗	✗	0.2	0.2
Cutmix Prob.	✗	✗	✗	0.8	0.8

Table 7: Implementation details on **ImageNet-100**.

Hyperparameters	**Models**					
	WONN	***AKOrN***^{attn}	**ResNet-18**	**ResNet-50**	**ViT-S-16**	**ViT-B-16**
Batch Size	64	16	128	128	256	256
Optimizer	AdamW	Adam	AdamW	AdamW	AdamW	AdamW
Weight Decay	0.0001	–	0.0001	0.0001	0.05	0.05
LR	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	$2/3 \cdot 10^{-4}$
LR Decay	✗	✗	✗	✗	cosine	cosine
Min LR	✗	✗	✗	✗	0	0
Warmup Epochs	✗	✗	✗	✗	10	10
Epochs	200	200	200	200	300	300
Dynamics Layers (L)	6	3	–	–	–	–
Dynamics Time Step (T)	3	3	–	–	–	–
Group Size (N)	2	–	–	–	–	–
Oscillator Dimension (n)	–	4	–	–	–	–
Input Patch Size	4	4	–	–	16	16
GPUs	1×H100	1×H100	1×H100	1×H100	1×H100	1×H100
Dropout	✗	✗	✗	✗	✗	✗
Label Smoothing	✗	✗	✗	✗	0.1	0.1
Mixup Alpha	✗	✗	✗	✗	0.8	0.8
Cutmix Alpha	✗	✗	✗	✗	1.0	1.0
Mixup Prob.	✗	✗	✗	✗	0.2	0.2
Cutmix Prob.	✗	✗	✗	✗	0.8	0.8

Table 8: Implementation details on **ImageNet-1K**.

Hyperparameters	**Models**					
	WONN	***AKOrN***^{attn}	**ResNet-18**	**ResNet-50**	**ViT-S-16**	**ViT-B-16**
Batch Size	512	512	128	128	1024	1024
Optimizer	AdamW	AdamW	AdamW	AdamW	AdamW	AdamW
Weight Decay	0.001	0.001	0.001	0.001	0.05	0.05
LR	$7.5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$5 \cdot 10^{-4}$	$3 \cdot 10^{-4}$	$2/3 \cdot 10^{-4}$
LR Decay	cosine	cosine	cosine	cosine	cosine	cosine
Min LR	$1 \cdot 10^{-6}$	$1 \cdot 10^{-6}$	$1 \cdot 10^{-6}$	$1 \cdot 10^{-6}$	0	0
Warmup Epochs	10	10	10	10	10	10
Epochs	300	200	200	200	300	300
Dynamics Layers (L)	6	3	–	–	–	–
Dynamics Time Step (T)	3	3	–	–	–	–
Group Size (N)	2	–	–	–	–	–
Oscillator Dimension (n)	–	4	–	–	–	–
Input Patch Size	7	7	–	–	16	16
GPUs	4×H100	8×H200	1×H100	1×H100	4×H100	4×H100
GPU hours	750	950	–	–	–	–
Dropout	✗	✗	✗	✗	✗	✗
Mixup Alpha	–	–	–	–	0.8	0.8
Cutmix Alpha	1.0	1.0	1.0	1.0	1.0	1.0
Label Smoothing	0.1	0.1	0.1	0.1	0.1	0.1
Mixup Prob.	0.0	0.0	0.0	0.0	0.2	0.2
Cutmix Prob.	0.8	0.8	0.8	0.8	0.8	0.8

C.1.2 Solving Maze-hard

For the Maze-hard task, we train the two models *WONN* and *AKOrN* under the same training protocol, with implementation details summarized in Table 9. Results for the remaining baselines, including large language models and recurrent models, are taken from [44, 16].

Table 9: Implementation details on **Maze-hard**.

Hyperparameters	**Models**	
	WONN	***AKOrN***^{attn}
Batch Size	64	64
Channel	128	256
Optimizer	Adam	Adam
Weight Decay	–	–
LR	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$
LR Decay	–	–
Min LR	–	–
Warmup Epochs	–	–
Epochs	9000	9000
Dynamics Layers (L)	1	1
Dynamics Time Step (T)	24	24
Group Size (N)	1	–
GPUs	1×H100	1×H100

C.1.3 Solving Sudoku

For the Sudoku task, we train *WONN* following the same training protocol as *AKOrN*. The detailed configuration is summarized in Table 10. The *HRM* results are obtained from our reimplementation, while results for the remaining baselines are taken from prior work [45, 30, 26].

Table 10: Implementation details on **Sudoku**.

Hyperparameters	**Models**		
	WONN	***AKOrN***^{attn}	***HRM***
Batch Size	100	100	384
Channel	256	512	512
Optimizer	Adam	Adam	Adam-atan2
Weight Decay	–	–	1.0
LR	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$1 \cdot 10^{-4}$
LR Decay	–	–	–
Min LR	–	–	–
Warmup Epochs	–	–	2000 steps
Epochs	100	100	2000
Dynamics Layers (L)	1	1	$H/L = 4/4$
Dynamics Time Step (T)	16	16	–
Group Size (N)	1	–	–
GPUs	1×H100	1×H100	1×H100

C.2 Additional Results and Ablations on Image Recognition

Here we provide additional experimental results for image recognition benchmarks, including supplementary evaluations on ImageNet-100 (Table 11) and CIFAR-100 (Table 12). We further conduct several ablation studies on CIFAR-100 to analyze the contribution of key architectural and dynamical components of *WONN*. Throughout this section, L denotes the number of Winfree dynamics layers, T denotes the number of recurrent dynamics steps per layer, and N denotes the group size. Tables 11 and 12 summarize the ablation results on ImageNet-100 and CIFAR-100, respectively.

Table 11: Additional image classification results on **ImageNet-100**.

Models	**Accuracy (%)**	**# Parameters**
AKOrN^{attn} (input patch size = 7)	75.12	4.63M
AKOrN^{attn} (input patch size = 4)	80.08	4.62M
$S(\theta_i)$ and $I(\theta_i)$ as MLPs, input patch size = 8		
WONN ($\mathrm{Ch} = 256 \to 256$)	82.08	12.09M
$S(\theta_i)$ and $I(\theta_i)$ as MLPs, input patch size = 7		
WONN ($\mathrm{Ch} = 128 \to 128$)	78.78	3.11M
WONN ($\mathrm{Ch} = 64 \to 256$)	79.12	7.57M
WONN ($\mathrm{Ch} = 256 \to 256$)	80.08	12.08M
$S(\theta_i)$ and $I(\theta_i)$ as MLPs, input patch size = 4		
WONN ($\mathrm{Ch} = 128 \to 128$)	81.50	3.09M
WONN ($\mathrm{Ch} = 64 \to 256$)	82.04	7.56M
WONN ($\mathrm{Ch} = 256 \to 256$)	82.88	12.05M
$S(\theta_i)$ and $I(\theta_i)$ as trigonometric functions, input patch size = 7		
WONN ($\mathrm{Ch} = 128 \to 128$)	77.92	3.01M
WONN ($\mathrm{Ch} = 64 \to 256$)	78.78	7.44M
WONN ($\mathrm{Ch} = 256 \to 256$)	80.34	11.89M
$S(\theta_i)$ and $I(\theta_i)$ as trigonometric functions, input patch size = 4		
WONN ($\mathrm{Ch} = 128 \to 128$)	81.76	3.00M
WONN (($\mathrm{Ch} = 64 \to 256$)	82.56	7.43M
WONN ($\mathrm{Ch} = 256 \to 256$)	82.22	11.87M

Table 12: Additional image classification results on **CIFAR-100**.

Models	**Accuracy (%)**	**# Parameters**
$S(\theta_i)\&I(\theta_i)$ as MLPs, $L = 6$, $T = 3$, $N = 2$		
WONN (Ch $= 128 \rightarrow 128$)	73.77$_{\pm 0.40}$	3.09M
WONN (Ch $= 64 \rightarrow 256$)	75.12$_{\pm 0.49}$	7.56M
WONN (Ch $= 256 \rightarrow 256$)	76.20$_{\pm 0.45}$	12.04M
$S(\theta_i)\&I(\theta_i)$ as trigonometric functions, $L = 6$, $T = 3$, $N = 2$		
WONN (Ch $= 128 \rightarrow 128$)	74.48$_{\pm 0.16}$	3.00M
WONN (Ch $= 64 \rightarrow 256$)	75.81$_{\pm 0.33}$	7.43M
WONN (Ch $= 256 \rightarrow 256$)	76.17$_{\pm 0.52}$	11.86M
$S(\theta_i)\&I(\theta_i)$ as MLPs, Ch=256, local (conv) and globle (attn) coupling		
WONN^{attn} (L1T3N2)	66.14	1.65M
WONN^{attn} (L6T1N2)	74.53	12.04M
WONN^{attn} (L6T3N1)	75.93	11.82M
WONN^{attn} (L6T3N2)	76.20	12.04M
WONN^{conv} (L6T3N2)	75.63	18.72M
WONN^{attn} (L6T3N4)	75.13	15.14M
WONN^{attn} (L6T6N2)	76.40	12.04M
WONN^{attn} (L9T3N2)	76.77	18.28M
$S(\theta_i)\&I(\theta_i)$ as trigonometric functions, Ch=256, local (conv) and globle (attn) coupling		
WONN^{attn} (L1T3N2)	66.07	1.62M
WONN^{attn} (L6T1N2)	74.58	11.86M
WONN^{attn} (L6T3N2)	76.17	11.86M
WONN^{attn} (L6T3N1)	76.94	11.79M
WONN^{conv} (L6T3N2)	76.27	18.54M
WONN^{attn} (L6T3N4)	76.24	12.67M

Impact of Patch Size. Table 11 studies the input patch size on ImageNet-100. For both MLPs-based and trigonometric interaction functions, a smaller patch size generally improves performance, suggesting that finer spatial tokenization might provides more effective oscillatory representations.

Impact of Interaction Parameterization. We compare two choices for the sensitivity and influence functions, namely learnable MLPs and fixed trigonometric functions. Both variants achieve strong performance, indicating that the benefit of *WONN* mainly comes from the oscillatory dynamical structure rather than a specific parameterization. The trigonometric form provides a stronger oscillator-inspired inductive bias and tends to outperfrom in small datasets, while the learnable MLP parameterization offers greater flexibility, which may be beneficial for scaling to larger and more complex datasets.

Impact of Depth (L). Increasing the number of Winfree dynamics layers generally improves performance. For example, moving from shallow layers to $L = 6$ substantially improves CIFAR-100 accuracy, showing that stacked dynamical refinement is important for representation learning.

Impact of Dynamics Steps (T). Increasing the number of recurrent dynamics steps also tends to improve performance, but at the cost of additional computation. This suggests a trade-off between the evolution horizon of synchronization dynamics and computational efficiency.

Impact of Group Size (N). The effect of group size depends on the interaction form. MLP-based interactions perform best at $N = 2$, while trigonometric interactions benefit from a different grouping choice, with $N = 1$ outperforming $N = 2$ among grouped variants. This suggests that selecting an appropriate group size can enhance WONN performance by matching the interaction scale to the parameterization of the interaction function.

Impact of Coupling Type (conv vs. attn). We compare global attentive coupling with local convolutional coupling under both MLP-based and trigonometric interactions. Although convolutional coupling achieves comparable accuracy, it introduces substantially more parameters in our implementation. Attentive coupling therefore offers a more efficient mechanism for coordinating oscillators, and its better parameter efficiency may make it more favorable for scaling WONN to larger and more complex datasets.

C.3 Fixed and Learnable Components

Table 13 distinguishes fixed and learnable components of *WONN* under the trigonometric and MLP-based parameterizations.

Table 13: Fixed and learnable components of *WONN*. "Fixed" denotes operations without trainable parameters or manually specified hyperparameters.

Component	Trigonometric	MLP-based
Point-wise sensitivity $S(\theta)$	Fixed $\cos\theta$	Learnable MLP
Point-wise influence $I(\theta)$	Fixed $\sin\theta$	Learnable MLP
Coupling (attention or convolution)	Learnable	Learnable
Layer-transition mappings	Learnable	Learnable
Input embedding and output head	Learnable	Learnable
Hyperparameters γ, L, T_l, and N	Fixed	Fixed

C.4 Solving Maze-hard

Empirically, we observe that the interaction energy E_{int} consistently decreases as the number of time steps T increases, serving as a reliable metric for the degree of synchronization (as detailed in Sec. 2.3). Leveraging this property, we introduce the energy-based voting mechanism during inference to enhance model performance. Specifically, we appropriately extend the evaluation time steps T and perform K independent trials with different random initializations of initial phases. The prediction corresponding to the lowest final interaction energy E_{int} is ultimately selected as the optimal solution, which empirically boosts the performance.

Table 14: Solving Maze-hard results

Models	Accuracy (%)	# Parameters
Other synchrony-based model		
AKOrN^{attn}(L1T16)	32.3	1M
AKOrN^{attn}(L1T24)	36.2	1M
$S(\theta_i)$ & $I(\theta_i)$ as trigonometric functions, L=1, T=24		
WONN	76.2	**0.396M**
WONN ($T_{eval} = 25$, K=32)	80.1	**0.396M**

C.5 Computational Cost and Memory Usage

We report additional measurements of training latency, single-sample inference latency, and peak GPU memory on Sudoku and Maze-hard. All measurements are conducted on a single NVIDIA H100 using the same numerical precision and timing protocol as the CIFAR-10 measurements in Sec. 3.5. Training latency is measured with batch size 32.

Sudoku. We compare *WONN* with the corresponding *AKOrN* reasoning model in Table 15.

Table 15: Training and inference costs on Sudoku.

Model	Params.	Latency (ms)		Peak memory (GiB)	
		Train/batch	Infer/sample	Train	Infer
WONN	1.583M	**28.523** $\pm$ 0.106	**1.820** $\pm$ 0.018	0.336	0.042
AKOrN	2.982M	37.272 $\pm$ 0.044	2.247 $\pm$ 0.017	0.780	0.020

On Sudoku, *WONN* uses fewer parameters and achieves lower training and inference latency and lower training memory than *AKOrN*.

Maze-hard. For *WONN* and *AKOrN*, the reported inference latency corresponds to a single trajectory ($K = 1$). *HRM* and *TRM* use their standard deterministic inference procedures. Table 16 reports the results.

Table 16: Training and inference costs on Maze-hard.

Model	Params.	Latency (ms)		Peak memory (GiB)	
		Train/batch	Infer/sample	Train	Infer
WONN	0.396M	**40.683** $\pm$ 0.494	**3.099** $\pm$ 0.067	2.021	0.036
AKOrN	0.957M	62.927 $\pm$ 0.003	**2.990** $\pm$ 0.035	6.146	0.011
HRM†	27.270M	51.347 $\pm$ 0.205	50.279 $\pm$ 1.233	5.081	0.148
TRM†	6.823M	43.214 $\pm$ 0.152	48.093 $\pm$ 0.223	5.706	0.087

† The reported *HRM* and *TRM* training latencies correspond to only one Adaptive Computation Time (ACT) step. A complete update executes a data-dependent number of ACT steps, up to 16, whereas the *WONN* and *AKOrN* measurements correspond to complete training updates.

On Maze-hard, *WONN* completes a full training update faster than *AKOrN* and even faster than a single ACT step of *HRM* or *TRM*. It also has the lowest measured training memory usage among these models. At inference with $K = 1$, *WONN* is comparable to *AKOrN* and approximately $\mathbf{16.2}\times$ **faster than *HRM* and** $\mathbf{15.5}\times$ **faster than *TRM***.

Energy-voting cost. For the $K = 32$ energy-voting setting used to obtain 80.1% Maze-hard accuracy, we additionally measure serial and parallel execution, as reported in Table 17.

Table 17: Inference costs of energy voting on Maze-hard with $K = 32$ trajectories. Speedup is relative to serial execution.

Execution mode	Latency (ms/maze)	Speedup	Peak memory (GiB)
Serial	97.197 $\pm$ 1.087	1.000$\times$	0.038
Parallel	**13.158** $\pm$ 0.132	7.387$\times$	0.104

Parallel execution reduces latency by a factor of 7.387. Even with $K = 32$ trajectories, parallel *WONN* inference remains approximately $\mathbf{3.8}\times$ **faster than *HRM* and** $\mathbf{3.7}\times$ **faster than *TRM*** under their standard inference procedures.

D Energy Structure and Topological Obstruction

We clarify the energy structure of the trigonometric Winfree dynamics used in *WONN*. For simplicity, we analyze the continuous-time phase dynamics

$$\dot{\theta}_i = \omega_i + \gamma \cos\theta_i \sum_j c_{ij} \sin\theta_j, \qquad \theta_i \in \mathbb{T} := \mathbb{R}/(2\pi\mathbb{Z}), \tag{11}$$

where ω_i is the natural frequency, $\gamma > 0$ is the interaction strength, and c_{ij} is the coupling coefficient from oscillator j to oscillator i.

Absence of a global potential in the presence of natural frequencies. We first show that the natural-frequency term prevents the full dynamics from being written as a globally defined gradient flow on the torus. Consider the pure frequency dynamics

$$\dot{\theta}_i = \omega_i. \tag{12}$$

If this vector field were the negative gradient of a globally defined scalar potential $E : \mathbb{T}^d \to \mathbb{R}$, then in local phase coordinates it would satisfy

$$-\frac{\partial E}{\partial \theta_i} = \omega_i, \qquad \text{or equivalently} \qquad \frac{\partial E}{\partial \theta_i} = -\omega_i. \tag{13}$$

Locally, this implies

$$E(\Theta) = -\sum_i \omega_i \theta_i + \text{const.} \tag{14}$$

However, θ_i is not a globally single-valued function on $\mathbb{T}$. Since θ_i and $\theta_i + 2\pi$ represent the same point on the circle, a globally defined potential must be invariant under the identification

$$\theta_i \sim \theta_i + 2\pi.$$

But the local potential changes by

$$E(\Theta + 2\pi e_i) - E(\Theta) = -2\pi\omega_i. \tag{15}$$

Therefore, when $\omega_i \neq 0$, this local potential is not single-valued on the torus. Equivalently, the constant frequency drift has nonzero circulation along the fundamental cycle of the i-th circle:

$$\oint \omega_i \, d\theta_i = 2\pi\omega_i \neq 0. \tag{16}$$

Since the gradient of a globally defined scalar potential must have zero circulation along every closed loop, the natural-frequency term is a topological obstruction to a global potential on $\mathbb{T}^d$.

Interaction energy. Although the full dynamics do not admit a globally defined potential in the presence of natural frequencies, the trigonometric interaction term admits a natural energy under symmetric coupling. Assume that the coupling matrix is fixed and symmetric:

$$c_{ij} = c_{ji}. \tag{17}$$

Define the interaction energy

$$E_{\mathrm{int}}(\Theta) = -\frac{1}{2}\sum_{i,j} c_{ij} \sin\theta_i \sin\theta_j. \tag{18}$$

Then

$$\frac{\partial E_{\mathrm{int}}}{\partial \theta_i} = -\cos\theta_i \sum_j c_{ij} \sin\theta_j. \tag{19}$$

Therefore, the trigonometric interaction term can be written as

$$\cos\theta_i \sum_j c_{ij} \sin\theta_j = -\frac{\partial E_{\mathrm{int}}}{\partial \theta_i}. \tag{20}$$

Substituting this into Eq. (11), the full dynamics can be decomposed as

$$\dot{\theta}_i = \omega_i - \gamma \frac{\partial E_{\mathrm{int}}}{\partial \theta_i}. \tag{21}$$

Thus, the system is a drift-plus-gradient system rather than a pure gradient flow.

The same structure can also be expressed geometrically on the embedded circle. Let

$$x_i = \begin{pmatrix} \cos\theta_i \\ \sin\theta_i \end{pmatrix} \in S^1, \qquad T_i = \frac{\partial x_i}{\partial \theta_i} = \begin{pmatrix} -\sin\theta_i \\ \cos\theta_i \end{pmatrix}. \tag{22}$$

Here T_i is the unit tangent vector at x_i, satisfying $x_i^\top T_i = 0$. Since $\dot{x}_i = T_i \dot{\theta}_i$, Eq. (21) becomes

$$\dot{x}_i = \omega_i T_i - \gamma \, \mathrm{grad}^{S^1}_{x_i} E_{\mathrm{int}}. \tag{23}$$

The first term is the natural-frequency drift along the circle, while the second term is the Riemannian gradient flow induced by the interaction energy.

Why full dynamics does not admit a Lyapunov function. Although E_{int} captures the interaction part of the dynamics, it is not in general a Lyapunov function for the full system when $\omega \neq 0$. Indeed, along trajectories of Eq. (21),

$$\frac{dE_{\mathrm{int}}}{dt} = \sum_i \frac{\partial E_{\mathrm{int}}}{\partial \theta_i} \dot{\theta}_i \tag{24}$$

$$= \sum_i \omega_i \frac{\partial E_{\mathrm{int}}}{\partial \theta_i} - \gamma \sum_i \left(\frac{\partial E_{\mathrm{int}}}{\partial \theta_i} \right)^2. \tag{25}$$

Equivalently,

$$\frac{dE_{\mathrm{int}}}{dt} = \langle \nabla_\Theta E_{\mathrm{int}}, \omega \rangle - \gamma \, \| \nabla_\Theta E_{\mathrm{int}} \|_2^2 . \tag{26}$$

The second term is non-positive, but the first term is not sign-definite. Consequently, E_{int} is not guaranteed to decrease along the full dynamics, and therefore it is not a Lyapunov function for the system with nonzero natural frequencies.

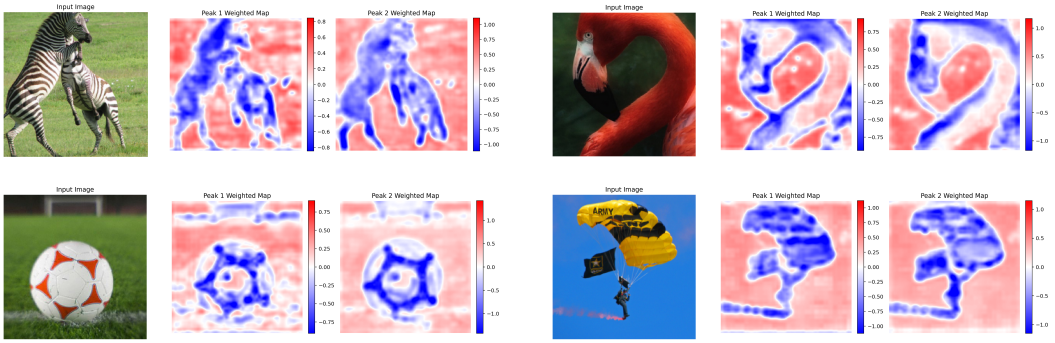


Figure 6: Additional qualitative visualizations of two-peak distributions on image recognition.

Lyapunov structure in the zero-frequency symmetric-coupling regime. If the natural-frequency term is removed, *i.e.,*

$$\omega_i = 0 \qquad \text{for all } i, \tag{27}$$

and the coupling matrix is fixed and symmetric, then Eq. (21) reduces to

$$\dot{\theta}_i = -\gamma \frac{\partial E_{\text{int}}}{\partial \theta_i}. \tag{28}$$

Thus the dynamics are a gradient flow on the torus:

$$\dot{\Theta} = -\gamma \nabla_{\Theta} E_{\text{int}}(\Theta). \tag{29}$$

In this case,

$$\frac{dE_{\text{int}}}{dt} = \sum_i \frac{\partial E_{\text{int}}}{\partial \theta_i} \dot{\theta}_i \tag{30}$$

$$= -\gamma \sum_i \left(\frac{\partial E_{\text{int}}}{\partial \theta_i} \right)^2 \tag{31}$$

$$\leq 0. \tag{32}$$

Therefore, in the zero-frequency symmetric-coupling regime, E_{int} is a Lyapunov function for the trigonometric interaction dynamics. In particular, isolated local minima of E_{int} correspond to Lyapunov-stable synchronized states.

Implication for *WONN*. The full *WONN* architecture contains frequency states, discrete-time updates, layer transitions and non-symmetric or state-dependent interactions. Therefore, the Lyapunov structure above should not be interpreted as a global stability theorem for the full model. Nonetheless, E_{int} provides a principled interaction energy for the trigonometric Winfree core and serves as a practical diagnostic of phase alignment. In Maze-hard (Sec. 3.3), we use this diagnostic for energy-based test-time selection among multiple stochastic phase trajectories.

E Additional Visualization

E.1 Additional Visualization on Image Recognition

Here we provide additional qualitative visualizations for image recognition. As shown in Fig. 6, these examples exhibit similar patterns to those discussed in Sec. 3.2, further supporting the observation that the phase modes of *WONN* capture complementary visual structures.

In addition to the static weighted maps shown in Sec. 3.2, we further visualize their layer-wise evolution in Fig. 7. The figures show how the weighted maps associated with the two dominant phase peaks evolve across recurrent Winfree dynamics steps and network depth. We observe that the phase-weighted responses are repeatedly refreshed at each layer: early layers mainly capture

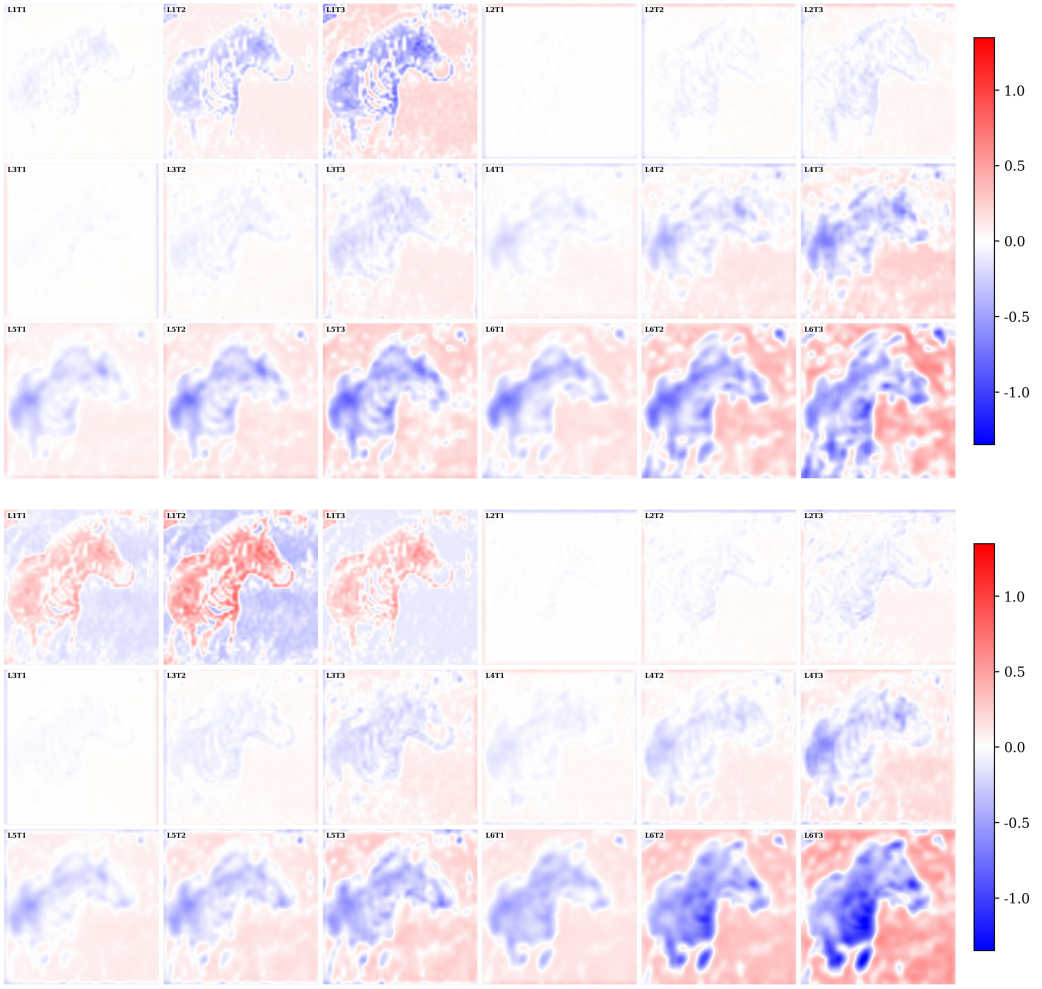


Figure 7: Layer-wise evolution of the weighted maps associated with the two dominant phase peaks in *WONN* on image recognition. **Top**: weighted map corresponding to the first dominant phase peak. **Bottom**: weighted map corresponding to the second dominant phase peak. Here L denotes the layer index and T denotes the Winfree dynamics step within that layer. Panels are arranged according to the actual forward trajectory, from $L1T1$ to $L6T3$. Across layers, the phase-weighted responses are progressively refreshed and reorganized, evolving from weak local activations toward more coherent global semantic structures through the recurrent synchronization dynamics.

weak and local visual structures, while deeper layers progressively organize these responses into more coherent global semantic patterns. This behavior suggests that the recurrent synchronization dynamics of *WONN* gradually refine and reorganize visual representations over depth and time.

E.2 Additional Visualization on Maze-hard

This subsection presents additional Maze-hard examples for qualitative analysis.

Fig. 8 illustrates the evolution of accuracy and interaction energy during training on the Maze-hard task. While the energy initially increases in the early stage of optimization, it subsequently decreases as training progresses, accompanied by a steady improvement in accuracy. This inverse correlation suggests that lower interaction energy is associated with more accurate solution states, indicating that the energy provides a useful proxy for solution quality in the learned synchronization dynamics.

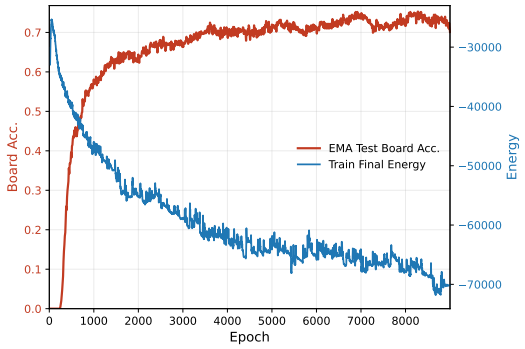


Figure 8: Accuracy–energy dynamics on Maze-hard. We report the test accuracy and the corresponding energy over training epochs. After a slight increase in the early stage, the energy gradually decreases as training proceeds, whereas the accuracy exhibits a steady improvement trend.

Fig. 10 shows the complete temporal evolution of the example discussed in Sec. 3.3, while Fig. 11 provides several additional instances. In the path prediction maps, black cells represent walls, red and green cells denote the start and goal, respectively, blue cells indicate the predicted path, and purple cells correspond to invalid wall-crossing predictions. Across different examples, we observe a consistent synchronization-driven dynamics: during the early stages, the network activates multiple candidate path fragments simultaneously rather than directly converging to a single solution. As the oscillatory dynamics evolve, these fragments progressively synchronize, compete, and merge into coherent global trajectories, while inconsistent or invalid paths are gradually suppressed. Eventually, the dynamics converge to the shortest valid path connecting the start and goal. These visualizations suggest that synchronization dynamics provide a very natural mechanism for coordinating distributed local candidates into globally consistent reasoning solutions.

Fig. 9 show the temporal evolution of *HRM* on the same example discussed in Sec. 3.3, including both path predictions and probability maps. We visualize the first 12 H-block updates and regard them as the first 12 evolution steps. The results suggest that *HRM* remains largely inactive during the early stages, then begins to generate irregular predictions around $T = 4$, and subsequently undergoes an abrupt transition around $T = 6$ that recovers most of the correct path. The remaining steps only introduce minor refinements. This abrupt, insight-like behavior contrasts sharply with the progressive path formation exhibited by *WONN*.

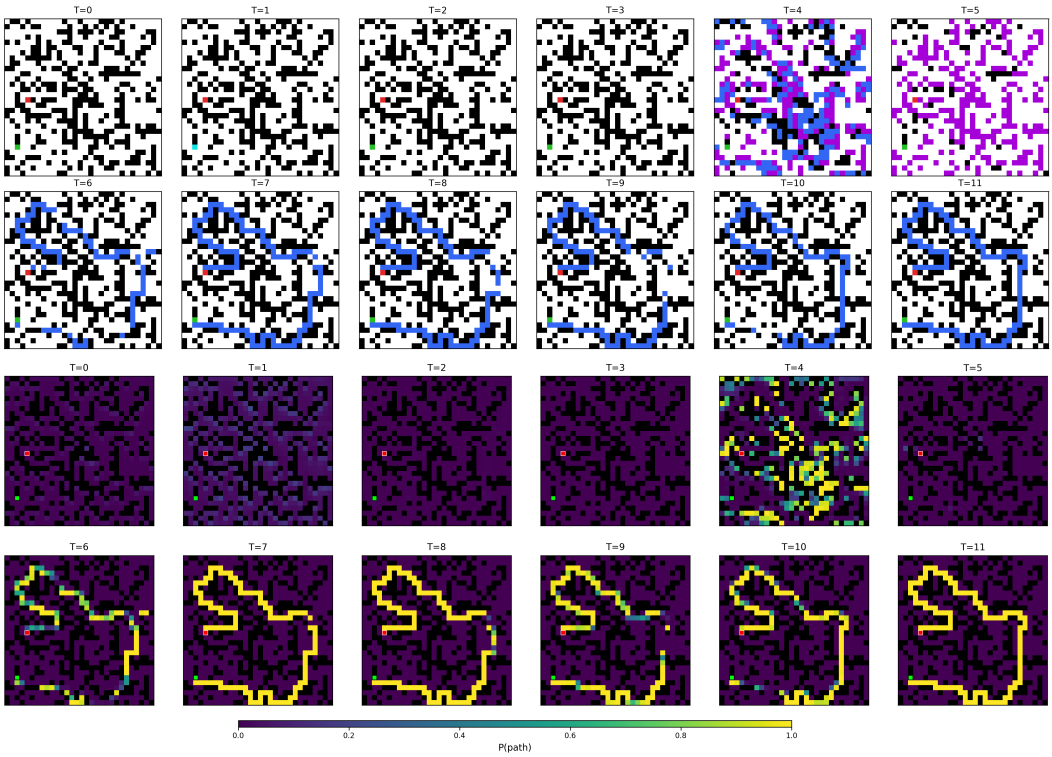


Figure 9: Partial temporal evolution of *HRM* on the example shown in Sec. 3.3. **Top:** discrete path predictions over time. **Bottom:** predicted path probability heatmaps over time. *HRM* remains mostly inactive during early H-block updates and then undergoes an abrupt, insight-like transition, while *WONN* progressively synchronizes diffuse candidate path fragments into a coherent valid path.

E.3 Additional Visualization on Sudoku

This section provides qualitative visualizations on the Sudoku task. Fig. 12 shows that *WONN* follows a dynamical solving process similar to that observed on Maze-hard. At early time steps, the model tends to perform a global synchronized exploration of possible assignments. During subsequent evolution, these predictions are gradually refined, and the model progressively converges to the correct solution.

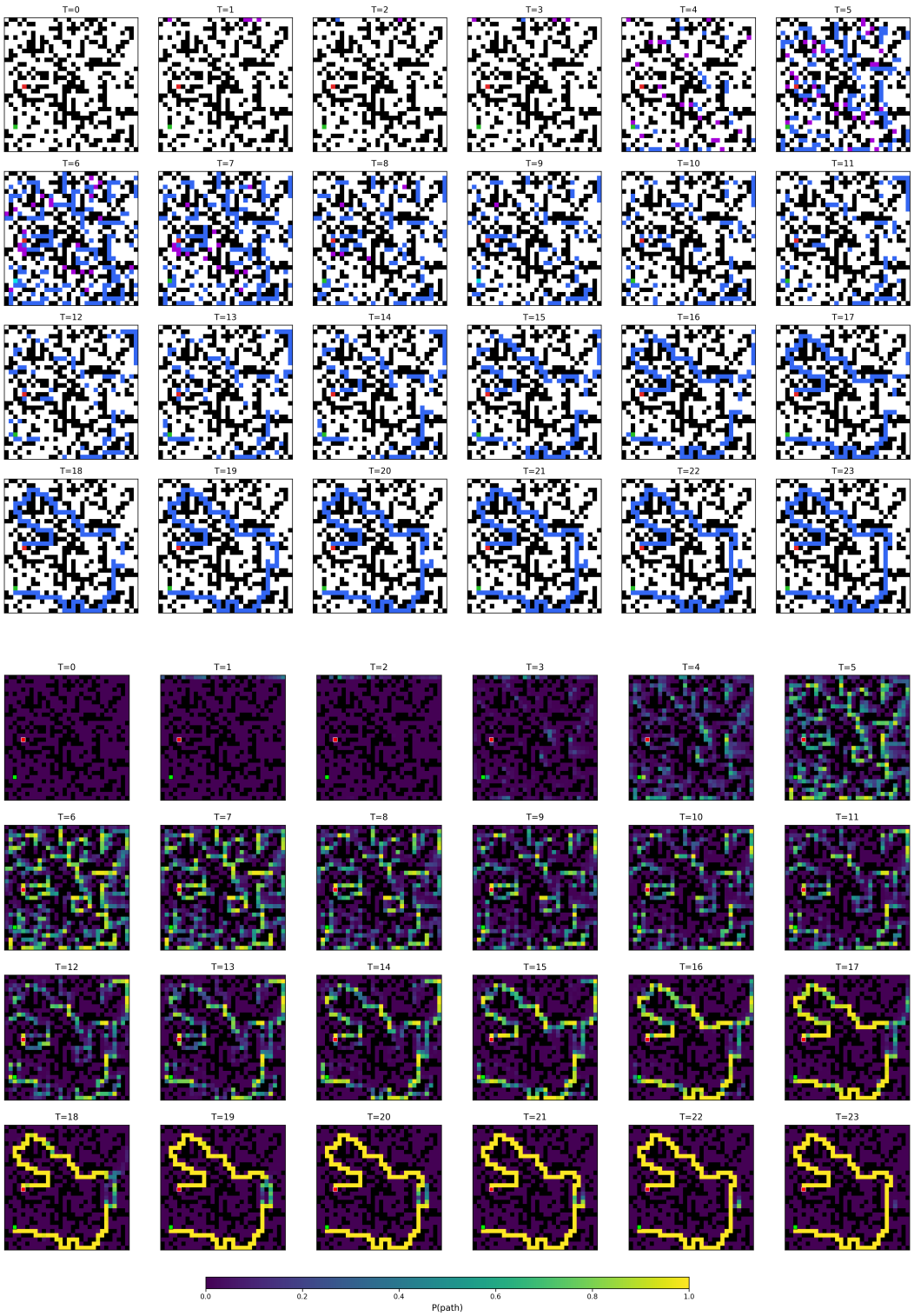
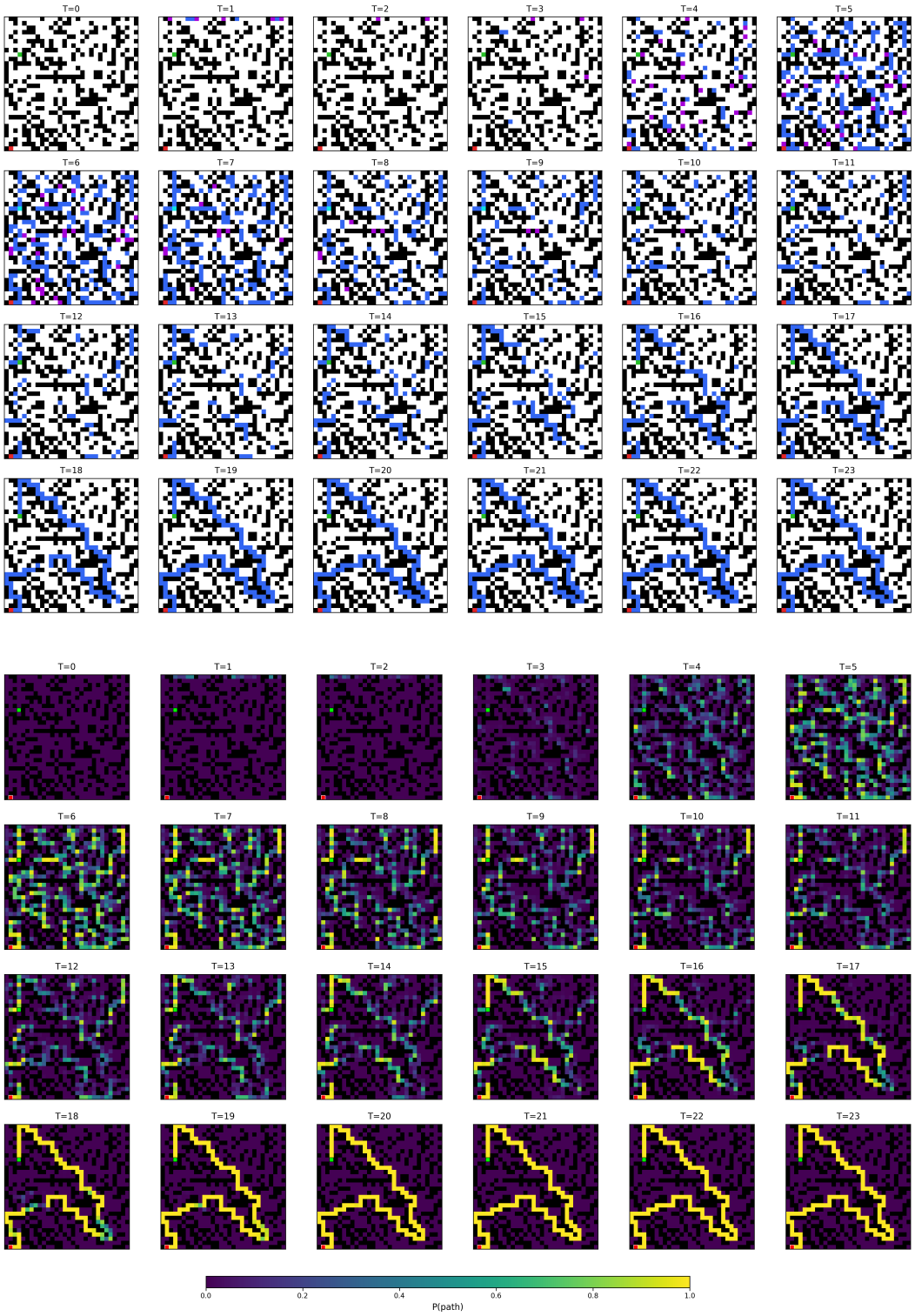
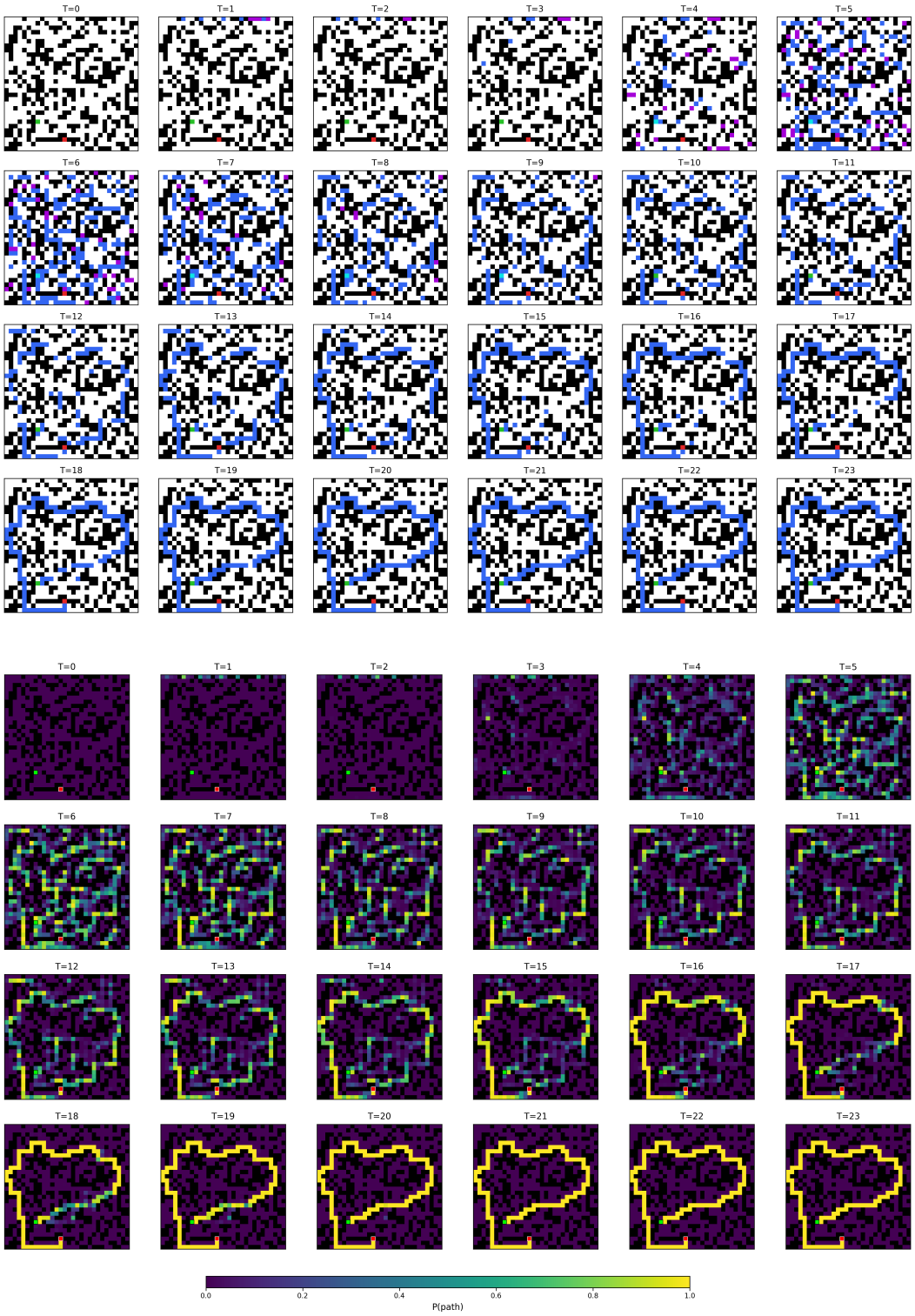


Figure 10: Complete temporal evolution of the example shown in Sec. 3.3. **Top:** discrete path predictions over time. **Bottom:** predicted path probability heatmaps over time.


T=0
T=1
T=2
T=3
T=4
T=5
T=6
T=7
T=8
T=9
T=10
T=11
T=12
T=13
T=14
T=15
T=16
T=17
T=18
T=19
T=20
T=21
T=22
T=23
0.0
0.2
0.4
0.6
0.8
1.0
P(path)


T=0
T=1
T=2
T=3
T=4
T=5
T=6
T=7
T=8
T=9
T=10
T=11
T=12
T=13
T=14
T=15
T=16
T=17
T=18
T=19
T=20
T=21
T=22
T=23
T=0
T=1
T=2
T=3
T=4
T=5
T=6
T=7
T=8
T=9
T=10
T=11
T=12
T=13
T=14
T=15
T=16
T=17
T=18
T=19
T=20
T=21
T=22
T=23
0.0
0.2
0.4
0.6
0.8
1.0
P(path)

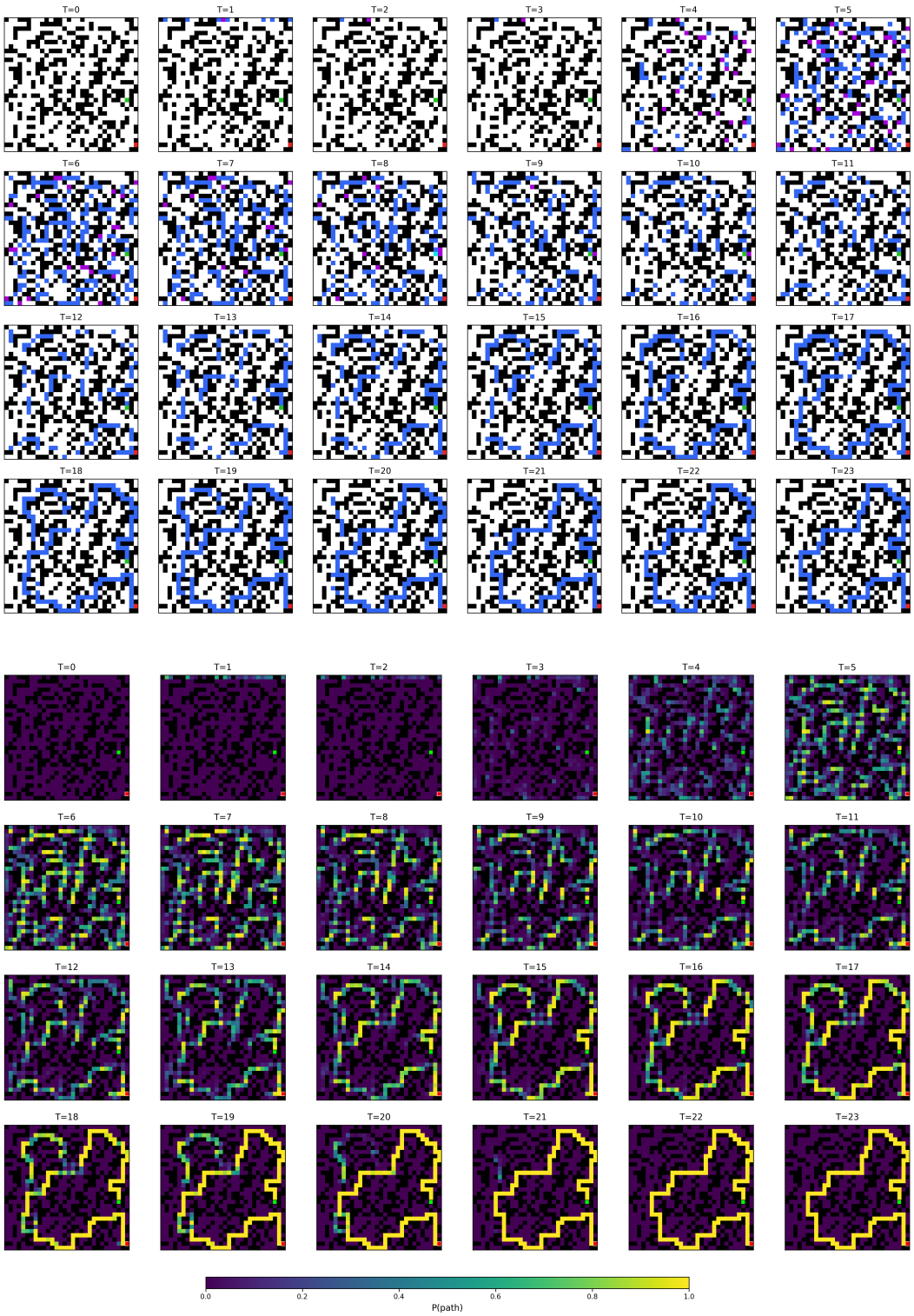


Figure 11: More Maze-hard pathfinding examples. Each pair shows the discrete path prediction and the corresponding path probability heatmap for one maze instance.

L1T1

4	4	4	4	4	3	4	4	4
3	6	1	4	5	5	5	3	4
4	5	7	3	4	6	5	5	4
4	3	4	4	3	4	5	5	5
1	4	5	3	4	3	4	4	5
4	3	3	8	5	4	3	3	3
6	5	5	5	3	3	4	3	4
4	4	8	3	5	4	3	4	5
4	5	3	4	5	5	4	3	4

L1T2

4	4	1	5	4	3	4	6	4
3	6	4	4	5	4	4	1	8
4	5	7	5	4	6	5	4	4
4	3	4	4	7	4	5	5	5
1	4	4	3	4	5	8	4	5
4	7	5	8	5	4	5	5	3
6	5	5	5	4	5	4	5	4
4	4	8	3	5	4	3	7	5
4	5	3	4	5	1	4	5	4

L1T3

4	4	1	5	4	3	4	6	4
3	6	4	4	5	4	4	1	8
4	3	7	1	4	6	5	3	4
5	3	4	4	7	4	1	5	3
1	4	9	3	4	5	8	4	5
4	7	3	8	5	4	9	5	5
6	5	1	4	3	5	4	5	4
4	4	8	1	4	9	3	7	5
5	3	3	4	5	1	4	3	4

L1T4

9	4	1	5	8	3	4	6	7
3	6	5	4	7	5	4	1	8
2	8	7	1	4	6	5	3	9
8	3	4	1	7	2	1	5	3
1	5	9	3	4	5	8	4	5
5	7	3	8	3	4	9	3	3
6	5	3	7	3	8	4	8	4
4	1	8	1	1	9	3	7	5
8	3	3	4	5	1	4	8	4

L1T5

9	4	1	5	8	3	4	6	7
3	6	5	2	8	7	4	1	8
2	8	7	1	4	6	5	3	9
8	3	4	1	7	2	1	5	3
1	4	9	3	6	5	8	4	7
5	7	3	8	3	4	9	3	5
6	5	3	7	3	8	1	8	4
4	1	8	1	6	9	3	7	5
7	3	3	4	5	1	2	8	4

L1T6

9	4	1	5	8	3	4	6	7
3	6	5	2	8	7	4	1	8
2	8	7	1	4	6	5	3	9
8	3	4	1	7	2	6	5	3
1	2	9	3	6	5	8	4	7
5	7	5	8	3	4	9	3	5
6	5	2	7	3	8	1	8	4
4	1	8	6	2	9	3	7	5
7	3	3	4	5	1	2	8	7

L1T7

9	4	1	5	8	3	7	6	2
3	6	5	2	9	7	4	1	8
2	8	7	1	4	6	5	3	9
8	3	4	1	7	2	6	5	3
1	2	9	3	6	5	8	4	7
5	7	3	8	1	4	9	2	1
6	5	2	7	3	8	1	8	4
4	1	8	6	2	9	3	7	5
7	3	3	4	5	1	2	8	6

L1T8

9	4	1	5	8	3	7	6	2
3	6	5	2	9	7	4	1	8
2	8	7	1	4	6	5	3	9
8	3	4	9	7	2	6	5	3
1	2	9	3	6	5	8	4	7
5	7	6	8	1	4	9	2	1
6	5	2	7	3	8	1	8	4
4	1	8	6	2	9	3	7	5
7	3	3	4	5	1	2	8	6

L1T9

9	4	1	5	8	3	7	6	2
3	6	5	2	9	7	4	1	8
2	8	7	1	4	6	5	3	9
8	3	4	9	7	2	6	5	3
1	2	9	3	6	5	8	4	7
5	7	6	8	1	4	9	2	3
6	5	2	7	3	8	1	8	4
4	1	8	6	2	9	3	7	5
7	9	3	4	5	1	2	8	6

L1T10

9	4	1	5	8	3	7	6	2
3	6	5	2	9	7	4	1	8
2	8	7	1	4	6	5	3	9
8	3	4	9	7	2	6	5	3
1	2	9	3	6	5	8	4	7
5	7	6	8	1	4	9	2	3
6	5	2	7	3	8	1	9	4
4	1	8	6	2	9	3	7	5
7	9	3	4	5	1	2	8	6

L1T11

9	4	1	5	8	3	7	6	2
3	6	5	2	9	7	4	1	8
2	8	7	1	4	6	5	3	9
8	3	4	9	7	2	6	5	1
1	2	9	3	6	5	8	4	7
5	7	6	8	1	4	9	2	3
6	5	2	7	3	8	1	9	4
4	1	8	6	2	9	3	7	5
7	9	3	4	5	1	2	8	6

L1T12

9	4	1	5	8	3	7	6	2
3	6	5	2	9	7	4	1	8
2	8	7	1	4	6	5	3	9
8	3	4	9	7	2	6	5	1
1	2	9	3	6	5	8	4	7
5	7	6	8	1	4	9	2	3
6	5	2	7	3	8	1	9	4
4	1	8	6	2	9	3	7	5
7	9	3	4	5	1	2	8	6

L1T13

9	4	1	5	8	3	7	6	2
3	6	5	2	9	7	4	1	8
2	8	7	1	4	6	5	3	9
8	3	4	9	7	2	6	5	1
1	2	9	3	6	5	8	4	7
5	7	6	8	1	4	9	2	3
6	5	2	7	3	8	1	9	4
4	1	8	6	2	9	3	7	5
7	9	3	4	5	1	2	8	6

L1T14

9	4	1	5	8	3	7	6	2
3	6	5	2	9	7	4	1	8
2	8	7	1	4	6	5	3	9
8	3	4	9	7	2	6	5	1
1	2	9	3	6	5	8	4	7
5	7	6	8	1	4	9	2	3
6	5	2	7	3	8	1	9	4
4	1	8	6	2	9	3	7	5
7	9	3	4	5	1	2	8	6

L1T15

9	4	1	5	8	3	7	6	2
3	6	5	2	9	7	4	1	8
2	8	7	1	4	6	5	3	9
8	3	4	9	7	2	6	5	1
1	2	9	3	6	5	8	4	7
5	7	6	8	1	4	9	2	3
6	5	2	7	3	8	1	9	4
4	1	8	6	2	9	3	7	5
7	9	3	4	5	1	2	8	6

L1T1

4	4	4	6	4	4	4	4	4
6	4	5	5	3	4	5	5	4
4	7	3	3	4	5	5	3	4
4	8	5	3	3	4	5	5	5
3	3	3	8	1	5	3	4	6
4	5	5	4	5	4	3	4	8
4	3	1	3	5	3	5	5	5
8	5	4	3	5	5	6	5	4
4	6	5	3	6	3	4	5	4

L1T2

4	4	4	6	4	4	1	4	4
6	4	5	4	5	4	5	5	7
4	7	4	3	4	5	5	4	4
4	8	5	4	3	4	4	5	5
3	5	5	8	1	4	5	4	6
7	5	5	4	6	4	3	4	8
4	4	1	5	5	3	8	5	5
8	5	4	4	4	5	6	5	4
5	6	5	4	7	3	4	5	4

L1T3

4	3	4	6	5	7	1	4	4
6	4	5	4	3	4	5	3	7
4	7	4	3	4	5	5	4	4
4	8	4	4	3	4	4	4	5
3	5	4	8	1	4	5	9	6
7	5	5	4	6	5	3	5	8
4	5	1	5	4	3	8	5	5
8	5	4	5	4	5	6	5	9
5	6	3	4	7	3	4	1	4

L1T4

4	3	9	6	5	7	1	8	4
6	4	5	7	8	4	5	3	7
2	7	4	3	8	8	5	4	4
4	8	4	7	3	9	2	3	5
3	4	4	8	1	7	5	9	6
7	5	7	4	6	5	3	5	8
4	5	1	5	3	7	8	5	4
8	4	4	5	4	3	6	5	9
5	6	8	1	7	3	4	3	1

L1T5

4	3	9	6	5	7	1	8	4
6	1	5	1	8	4	8	3	7
2	7	8	3	8	1	5	5	4
4	8	3	7	3	9	2	3	5
3	5	4	8	1	7	7	9	6
7	1	7	4	6	5	3	4	8
4	5	1	5	3	7	8	5	4
8	4	7	5	2	3	6	5	9
5	6	8	1	7	8	4	8	1

L1T6

4	3	9	6	5	7	1	8	3
6	1	5	1	8	4	9	3	7
2	7	8	3	8	1	5	5	4
1	8	6	7	3	9	2	3	5
3	5	4	8	1	7	7	9	6
7	9	2	4	6	5	3	1	8
4	5	1	5	3	3	8	7	4
8	4	7	5	2	3	6	5	9
5	6	8	9	7	8	4	8	1

L1T7

4	3	9	6	5	7	1	8	3
6	1	5	2	8	4	9	3	7
2	7	8	3	8	1	5	5	4
1	8	6	7	3	9	2	4	5
3	5	4	8	1	7	7	9	6
7	9	2	4	6	5	3	1	8
4	5	1	5	4	3	8	7	3
8	4	7	5	2	1	6	7	9
5	6	8	9	7	8	4	2	1

L1T8

4	3	9	6	5	7	1	8	3
6	1	5	2	8	4	9	8	7
2	7	8	3	8	1	5	6	4
1	8	6	7	3	9	2	4	5
3	5	4	8	1	2	7	9	6
7	9	2	4	6	5	3	1	8
4	2	1	5	4	6	8	7	3
8	4	7	5	2	3	6	7	9
5	6	3	9	7	8	4	2	1

L1T9

4	3	9	6	5	7	1	8	2
6	1	5	2	8	4	9	3	7
2	7	8	3	8	1	5	6	4
1	8	6	7	3	9	2	4	5
3	5	4	8	1	2	7	9	6
7	9	2	4	6	5	3	1	8
9	2	1	5	4	6	8	7	3
8	4	7	5	2	3	6	7	9
5	6	3	9	7	8	4	2	1

L1T10

4	3	9	6	5	7	1	8	2
6	1	5	2	8	4	9	3	7
2	7	8	3	9	1	5	6	4
1	8	6	7	3	9	2	4	5
3	5	4	8	1	2	7	9	6
7	9	2	4	6	5	3	1	8
9	2	1	5	4	6	8	7	3
8	4	7	1	2	3	6	5	9
5	6	3	9	7	8	4	2	1

L1T11

4	3	9	6	5	7	1	8	2
6	1	5	2	8	4	9	3	7
2	7	8	3	9	1	5	6	4
1	8	6	7	3	9	2	4	5
3	5	4	8	1	2	7	9	6
7	9	2	4	6	5	3	1	8
9	2	1	5	4	6	8	7	3
8	4	7	1	2	3	6	5	9
5	6	3	9	7	8	4	2	1

L1T12

4	3	9	6	5	7	1	8	2
6	1	5	2	8	4	9	3	7
2	7	8	3	9	1	5	6	4
1	8	6	7	3	9	2	4	5
3	5	4	8	1	2	7	9	6
7	9	2	4	6	5	3	1	8
9	2	1	5	4	6	8	7	3
8	4	7	1	2	3	6	5	9
5	6	3	9	7	8	4	2	1

L1T13

4	3	9	6	5	7	1	8	2
6	1	5	2	8	4	9	3	7
2	7	8	3	9	1	5	6	4
1	8	6	7	3	9	2	4	5
3	5	4	8	1	2	7	9	6
7	9	2	4	6	5	3	1	8
9	2	1	5	4	6	8	7	3
8	4	7	1	2	3	6	5	9
5	6	3	9	7	8	4	2	1

L1T14

4	3	9	6	5	7	1	8	2
6	1	5	2	8	4	9	3	7
2	7	8	3	9	1	5	6	4
1	8	6	7	3	9	2	4	5
3	5	4	8	1	2	7	9	6
7	9	2	4	6	5	3	1	8
9	2	1	5	4	6	8	7	3
8	4	7	1	2	3	6	5	9
5	6	3	9	7	8	4	2	1

L1T15

4	3	9	6	5	7	1	8	2
6	1	5	2	8	4	9	3	7
2	7	8	3	9	1	5	6	4
1	8	6	7	3	9	2	4	5
3	5	4	8	1	2	7	9	6
7	9	2	4	6	5	3	1	8
9	2	1	5	4	6	8	7	3
8	4	7	1	2	3	6	5	9
5	6	3	9	7	8	4	2	1

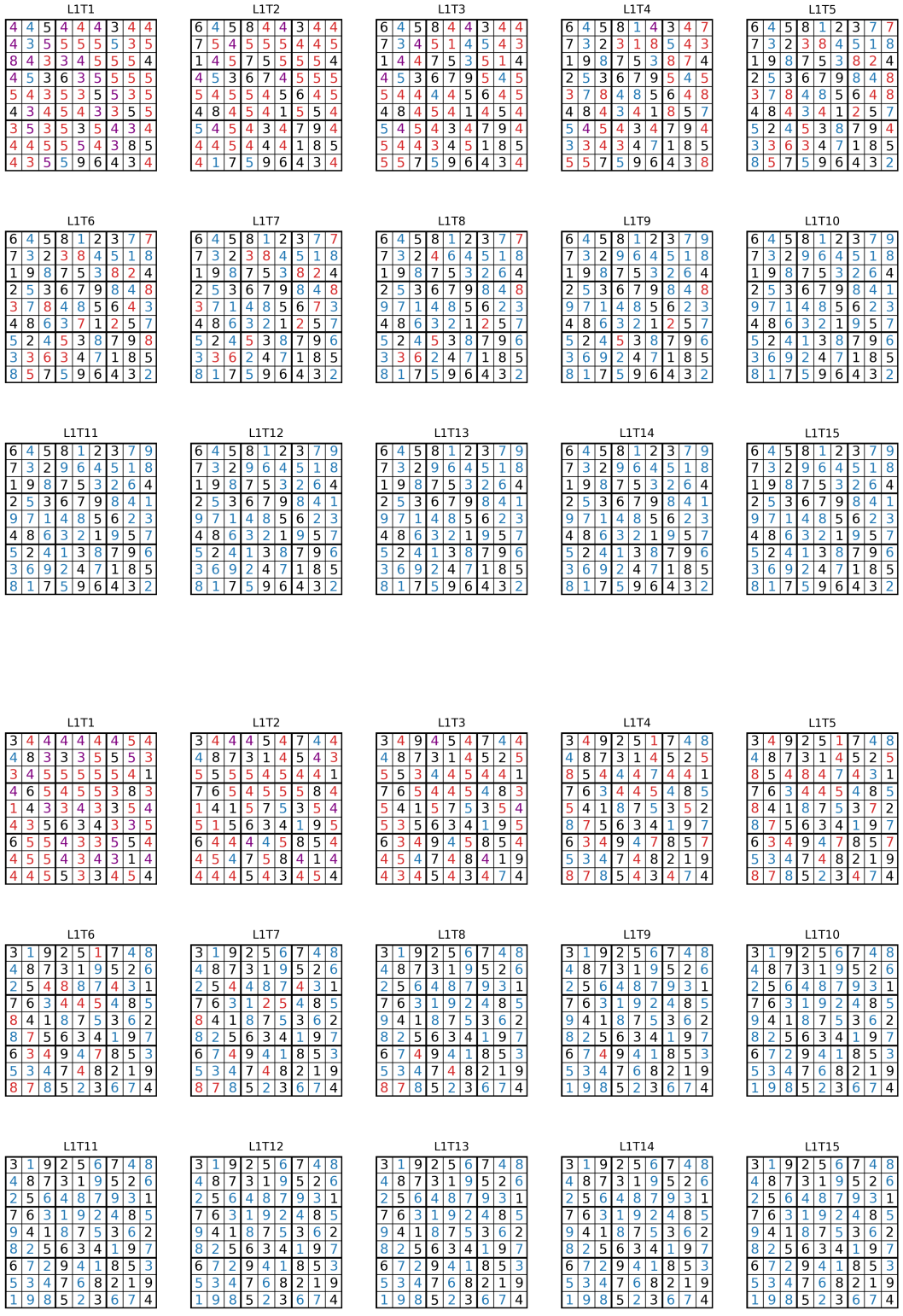


Figure 12: Temporal evolution of *WONN* on Sudoku. Early steps show global synchronized exploration over candidate digit assignments, while later steps progressively refine these predictions and converge to a correct globally consistent solution.

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: The main claims are validated in the method and experiment sections.

   Guidelines:

   - The answer [N/A] 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 [N/A] 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: We explicitly have a paragraph about limitations at the Appendix.

   Guidelines:

   - The answer [N/A] 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: [Yes]

Justification: The assumptions are listed and the proofs are in the Appendix.

Guidelines:

- The answer [N/A] 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: All the implementation details are released to guarantee the reproducibility.

   Guidelines:

   - The answer [N/A] 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: Our implementation is publicly available at `https://github.com/Jiawen-Dai/WONN`, together with installation instructions and scripts for training and evaluation. The datasets and experimental settings are described in Section 3 and Appendix C.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.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://neurips.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) necessary to understand the results?

Answer: [Yes]

Justification: The experimental settings are summarized in Section 3. Detailed training and evaluation configurations, including data splits, optimizers, and hyperparameters, are provided in Appendix C.

Guidelines:

- The answer [N/A] 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: We report the mean and standard deviation over multiple random seeds for experiments where repeated runs are conducted, including the CIFAR and Sudoku results. For larger-scale experiments with substantially higher computational cost, we report the corresponding single-run results together with detailed training configurations and compute resources.

Guidelines:

- The answer [N/A] 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: GPU types and counts are reported in Appendix C, along with training GPU-hours for ImageNet-1K. Training and inference latency and peak GPU memory for CIFAR-10, Sudoku, and Maze-hard are reported in Section 3.5 and Appendix C.5.

   Guidelines:

   - The answer [N/A] 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: The research respect the NeurIPS code of ethics.

   Guidelines:

   - The answer [N/A] 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: [N/A]

    Justification: There broader societal impact depends on the tasks and systems into which our models are deployed. We do not identify any unique concerns or societal implications beyond those already well established for research in deep learning.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] 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., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] 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: [Yes]

    Justification: All used existing assets get cited properly.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper does not involve any research with crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] 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: [N/A]

    Justification: The paper does not involve any research with crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] 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 rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The development of the core methodology does not involve any usages of LLMs

    Guidelines:

    - The answer [N/A] 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 in the NeurIPS handbook for what should or should not be described.