Shaping Sequence Attractor Schema in Recurrent Neural Networks

Zhikun Chu[1]
chukunzhi@outlook.com

Bo Hong[3]
hongbo@tsinghua.edu.cn

Xiaolong Zou[3,✉]
benzou@mail.bnu.edu.cn

Yuanyuan Mi[2,✉]
miyuanyuan@tsinghua.edu.cn

1, Bioengineering College,School of Medicine, Chongqing University.
2, Department of Psychological and Cognitive Sciences, Tsinghua University, Beijing, China.
3, Biomedical Engineering,School of Medicine, Tsinghua University.
✉: Corresponding authors.

Abstract

Sequence schemas are abstract, reusable knowledge structures that facilitate rapid adaptation and generalization in novel sequential tasks. In both animals and humans, shaping is an efficient way to acquire such schemas, particularly in complex sequential tasks. As a form of curriculum learning, shaping works by progressively advancing from simple subtasks to integrated full sequences, and ultimately enabling generalization across different task variations. Despite the importance of schemas in cognition and shaping in schema acquisition, the underlying neural dynamics at play remain poorly understood. To explore this, we train recurrent neural networks on an odor-sequence task using a shaping protocol inspired by well-established paradigms in experimental neuroscience. Our model provides the first systematic reproduction of key features of schema learning observed in the orbitofrontal cortex, including rapid adaptation to novel tasks, structured neural representation geometry, and progressive dimensionality compression during learning. Crucially, analysis of the trained RNN reveals that the learned schema is implemented through sequence attractors. These attractor dynamics emerge gradually through the shaping process: starting with isolated discrete attractors in simple tasks, evolving into linked sequences, and eventually abstracting into generalizable attractors that capture shared task structure. Moreover, applying our method to a keyword spotting task shows that shaping facilitates the rapid development of sequence attractor schemas, leading to enhanced learning efficiency. In summary, our work elucidates a novel attractor-based mechanism underlying schema representation and its evolution via shaping, offering new insights into the acquisition of abstract knowledge across biological and artificial intelligence.

1 Introduction

Imagine taking the subway in a new station: you can effortlessly anticipate the sequence of events - entering the station, purchasing a ticket, scanning it, waiting for the train, and boarding. This ability stems from an abstract knowledge structure encoded in your brain [1], which organizes the typical order and relationship among these events, commonly referred to as a schema. Schemas facilitate rapid learning [2, 3], flexible decision-making [4], and efficient generalization [5, 6, 7], forming the foundation of cognitive flexibility and generalization in both animal and human intelligence [3]. How-

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

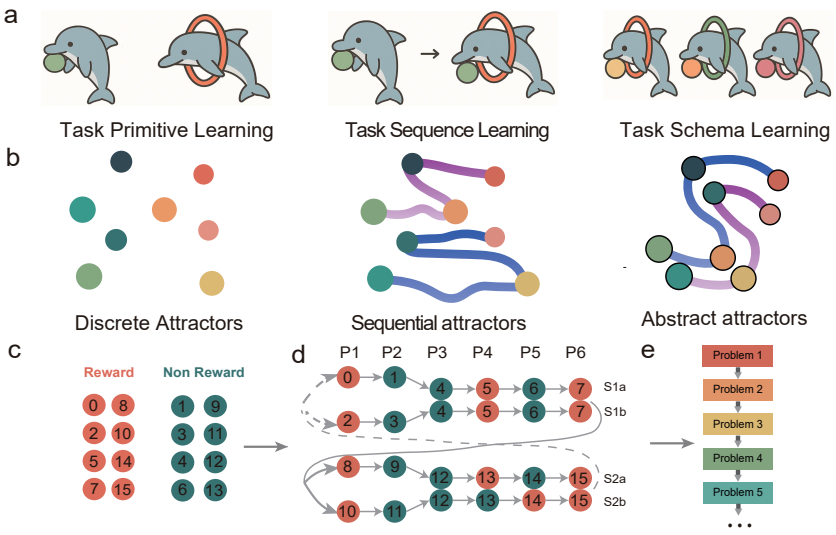

Figure 1: Schematic of learning sequence schema via Shaping. (a) A dolphin example. Task primitive learning: the dolphin learns basic actions independently, such as holding a ball and passing through a hoop. Task sequence learning: it learns to order these actions temporally - holding the ball first, then passing through the hoop. Task schema learning: exposure to different ball and hoop types allows the dolphin to develop generalizable schemas, enabling adaptation to similar tasks. (b) A neural dynamical hypothesis. Task primitive learning: Neural systems acquire basic attractor structures (e.g., discrete attractors). Task sequence learning: These discrete attractors are temporally linked into structured sequence attractors. Task schema learning: These sequence-specific attractors are compressed into an abstract schema. (c) Task primitive learning: An odor-reward association task is presented where 16 odors are linked to different rewards (red = rewarded, blue = non-rewarded). (d) Task sequence learning: Using the 16 odors, an odor-sequence task is constructed with two pairs of six-position sequences (S1a-S1b and S2a-S2b). Arrows show transitions between and within sequences. (e) Task schema learning: Five new problems with the same sequence structure but novel sets of 16 odors are introduced consecutively.

ever, despite their importance, the neural mechanisms underlying the representation and acquisition of schemas remain poorly understood.

For simple tasks, schemas can be learned through direct trial-and-error training on related experiences. However, for more complex sequential tasks, this approach often fails. Instead, animals and humans typically rely on shaping - a process that decomposes complex tasks into simpler subtasks learned incrementally [8, 9, 10]. As illustrated in Fig. 1a, teaching a dolphin to jump through a hoop while holding a ball exemplifies this approach. The animal first learns each basic skill independently (e.g., jumping through a hoop, holding a ball), then practices performing them in sequence, and ultimately generalizes the behavior across task variations (e.g., different ball colors or hoop types). Schema learning via shaping typically unfolds in three stages [4]: (1) task primitive learning, where basic action components are acquired; (2) task sequence learning, where these components are integrated into structured behavioral sequences; and (3) task schema learning, where abstract regularities are extracted across multiple structurally related tasks. While shaping has proven effective in practice, how it supports the learning and representation of task schemas remains unclear.

Schemas, primarily represented in the prefrontal cortex (PFC) [1, 3], are characterized by two key properties: low dimensionality and compositionality. The PFC is thought to support rich, low-dimensional attractor dynamics that underlie flexible behavior - for example, continuous attractor dynamics for sensory integration [11] and discrete attractors for sensorimotor transformations [12]. These low-dimensional dynamics have been proposed as neural representations of schemas [13] and are thought to underlie the learning-to-learn phenomena observed in primates [5]. Schemas are also compositional, allowing them to be reused and recombined to solve novel tasks [1]. For instance, rats can transfer learned spatial schemas from one context to novel memory tasks [2]. Moreover, recent

computational studies demonstrate that basic attractor dynamics, arising from multitask learning, can serve as primitive schemas that flexibly combine to facilitate new task learning [29, 15].

Building on these propoties, we propose a dynamical perspective on schema formation through shaping in complex sequential tasks (Fig. 1b), consisting of three stages: (1) Task primitive learning establishes basic attractors for individual task components, such as discrete attractors; (2) Task sequence learning integrates them into structured sequential dynamics, such as sequential attractors; (3) Task schema learning abstracts and compresses these patterns into unified low-dimensional representations that capture shared temporal and structural regularities.

Although the dynamic view of schema formation is compelling, it lacks a concrete computational model and direct comparison with neural data. In this work, we use recurrent neural networks (RNNs) to investigate how schema representations emerge and evolve through shaping. We validate our approach in an odor-sequence task widely used to study schema learning in the orbitofrontal cortex (OFC) [4]. Our main results and contributions are as follows.

(1) Our shaping-trained RNN systematically replicates key features of schema learning observed in rats' OFC ensembles, including faster learning on novel tasks, structured task representation geometry, and progressive dimensionality compression.

(2) The model reveals that schema representations emerge as low-dimensional sequence attractors, composed of discrete attractors linked by some low-dimensional manifolds, offering a mechanistic account of schema encoding and testable predictions for neuroscience.

(3) Dynamic analysis uncovers how attractor structures evolve during shaping: from isolated discrete attractors to integrated sequence attractors, to abstract, compact attractor structure - providing a novel dynamical mechanism of schema formation.

(4) Extending our sequence attractor-based shaping approach to the keyword spotting task shows improved learning efficiency, demonstrating its potential in complex, real-world tasks.

Together, our work advances the understanding of schema representation and its learning via shaping from a dynamical systems perspective, offering new insights into how abstract knowledge arises from experience through curriculum-like processes in both neuroscience and machine learning.

2 Experimental Background

To model schema learning under shaping conditions, we employ the odor-sequence task of schema evolution from Zhou et al. [4], a study designed to investigate this process in rats using a typical shaping protocol. In each task instance (termed a "problem"), 16 novel odors are randomly sampled and organized into two pairs of sequences (S1a – S1b and S2a – S2b), shown in Fig. 1d. In each of the six positions within a sequence, a single odor is presented, and the rat must decide if it signals a reward. Transitions among the sequences S1a, S1b, S2a, and S2b are selected at random. Although odor identities change in each new problem, the core sequence and reward structures remain the same. Thus, rats need to learn this invariant task structure for rapid adaptation. Zhou et al. adopted a shaping paradigm to train rats to learn the challenging task. (1) Task primitive learning: Initial training on a basic odor-reward association task with 16 randomly sampled odors (Fig. 1c). (2) Task sequence learning: Introduction of a structured sequence task using the same 16 odors and the predefined template (Fig. 1d), requiring the rat to link learned associations into temporal sequences. (3) Task schema learning: Introduction of five new problems successively, each with a novel set of 16 odors following the same task structure (Fig. 1e). The main findings of Zhou et al. are twofold: First, rats exhibit increasingly faster learning of new problems with more experience. Second, population activity in the OFC becomes progressively lower-dimensional and forms increasingly structured patterns that reflect the underlying task schema. However, the precise neural representation of schemas for odor-sequence tasks and the dynamics of their evolution through shaping are still unclear.

3 Methods

3.1 Model Structure

We employ an RNN model to demonstrate schema learning via shaping on the odor-sequence task. The RNN comprises three components: an input layer with M units, a recurrent layer with N units,

and an output layer with K units, as illustrated in Fig. 2a. At time t, given an input stimulus vector $I_m(t)$, the total input $x_n(t)$ to recurrent unit n evolves according to the following dynamics:

$$\tau \frac{dx_n(t)}{dt} = -x_n(t) + \sum_{j=1}^{N} W_{nj}^{\text{rec}} r_j(t) + \sum_{m=1}^{M} W_{nm}^{\text{in}} I_m(t), \tag{1}$$

here, W^{in} and W^{rec} represent the input-to-recurrent and recurrent-to-recurrent weight matrices, respectively. The activation of the unit n is computed as $r_n(t) = \tanh(x_n(t))$, and τ denotes the time constant.

The output $y_k(t)$ of unit k is given by:

$$y_k(t) = \sum_{n=1}^{N} W_{kn}^{\text{out}} r_n(t). \tag{2}$$

The output layer linearly reads out the RNN's activity to generate task-relevant predictions, including odor classification and reward prediction, depending on the shaping stage. The recurrent-to-output weight is denoted by W^{out}. The RNN is simulated using Euler's method of numerical integration.

3.2 Task Design and Shaping Procedure

In the odor sequence task, each odor stimulus is represented as a one-hot vector of dimension $M = 96$, with the non-zero entry set to 20 (see Appendix Sec.A for details). Each stimulus presentation lasts $T_{\text{s}} = 5$ steps. Following the shaping paradigm of Zhou et al. [4], the RNNs undergo training across three stages, detailed below.

Task Primitive Learning. In this stage, RNNs are trained on a simple prediction task that learn to associate individual odors with corresponding reward outcomes. Each 15-step trial includes a 5-step delay, a 5-step odor stimulus, and a 5-step delay, and is corrupted by Gaussian white noise with a mean of 0 and a variance of 1. The OFC flexibly encodes task-relevant variables, including odor rewards and identities [16, 17]. As both are required in this task, the model's output layer comprises $K = 18$ units: a 2-dimensional reward prediction vector $\hat{y}^{\text{reward}}(t)$ and a 16-dimensional odor classification vector $\hat{y}^{\text{class}}(t)$. For rewarded odors, $\hat{y}^{\text{reward}}(t) = [2, 0]$; for non-rewarded odors, $\hat{y}^{\text{reward}}(t) = [0, 2]$. Odor identity is encoded as a one-hot vector in $\hat{y}^{\text{class}}(t)$. Output units are required to be at zero before the odor stimulus appears, transition to target values during presentation, and remain there until it ends (Fig.S1). This target sequence mimics the ramping dynamics observed in cortical circuits during decision-making [18]. Finally, the loss function used during task primitive learning is defined as a regression loss:

$$L_1 = \frac{1}{2} \sum_{t=1}^{T} \left\| y_{0:2}(t) - \hat{y}^{\text{reward}}(t) \right\|^2 + \beta \sum_{t=1}^{T} \left\| y_{2:K}(t) - \hat{y}^{\text{class}}(t) \right\|^2. \tag{3}$$

The total loss is balanced between reward prediction and odor identity classification by a weighting factor β, which we set to 0.5 here.

Task Sequence Learning. During this stage, RNNs, having been pretrained in the task primitive learning phase, undergo further training in an odor-sequence problem. As described in Sec. 2 and Fig. 1d, the same 16 odors from the previous stage are organized into four sequences. Taking S1a as an example, each 90-step sequence trial comprises six odor stimuli presented at intervals sampled from a uniform distribution between 5 and 10 steps, mimicking experimental variability [4]. To simulate sensory noise, the entire input stream is corrupted with Gaussian noise (mean = 0, variance = 1). As odor identities in the sequence task become unreliable predictors of reward (illustrated by odor 13 indicating different rewards at P4 and P5), the network shifts its prediction target to only reward outcomes. Consequently, the output layer is simplified to $K = 2$ units, representing only the reward prediction vector $\hat{y}^{\text{reward}}(t)$, with the same values as in the task primitive learning setup. At the start of the sequence, all output targets are set to zero. Upon stimulus presentation, the outputs transition to their target values and hold them until the next odor appears, at which point they update accordingly. The loss function is defined as:

$$L_2 = \frac{1}{2} \sum_{t=1}^{T} \left\| y(t) - \hat{y}^{\text{reward}}(t) \right\|^2. \tag{4}$$

Task Schema Learning. As shown in Fig. 1e, the pretrained RNN from the previous stage is further trained on a series of odor-sequence problems sharing the same underlying structure. For each task, 16 new one-hot vectors with the active entry set to 20, different from those in the previous task, are selected from the 96-dimensional odor space. All other task parameters, including sequence duration and background noise level, remain consistent with those in task sequence learning. The network continues to predict only reward outcomes, with the loss defined as:

$$L_3 = \frac{1}{2} \sum_{t=1}^{T} \left\| y(t) - \hat{y}^{\text{reward}}(t) \right\|^2 . \tag{5}$$

Across all three stages, we use the Adam optimizer. See Appendix Sec.A for more training details.

4 Results

4.1 Learning-to-learn and Structured Task Representation Emerge in Shaping-Trained RNNs

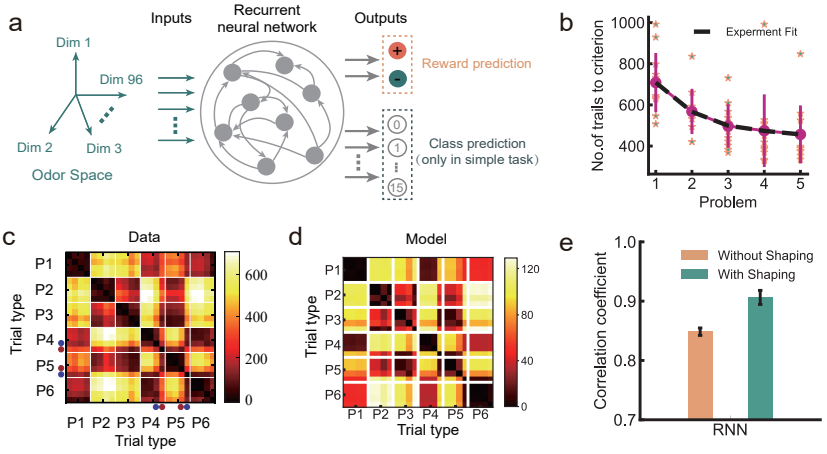

Figure 2: Behavior and representation in RNNs via shaping. (a) RNN architecture. A three-layer network (input, recurrent, output). The input is a 96-dimensional odor stimulus vector. The output layer predicts reward and odor class. (b) During task schema learning, the number of training trials for the shaped RNN to reach a performance criterion across consecutively presented problems (10 seeds per problem).The dashed line represents the polynomial fit. (c-d) Dissimilarity matrices. Representational dissimilarity matrices (RDM) of OFC ensembles(c) [4] and RNN hidden activities (d). Each matrix shows the distance between the neural/model representations of all pairs of trial types (24 total, 6 per sequence) across sequences and positions. (e) Pearson correlation between the experimental and model RDMs. Blue indicates RNNs trained with shaping, and pink indicates those without. Results are averaged over 10 seeds. See Appendix Sec.A for detailed parameters.

Schemas are proposed to encode abstract, generalizable knowledge, enabling faster learning in novel situations [2, 4]. To demonstrate this, we train an RNN using the shaping process described in Sec. 3.2. The RNN first learns a simple odor-reward association task until the training loss falls below 0.05, followed by an odor-sequence task using the same odors, again trained until the loss falls below 0.05. It is then exposed to a series of novel tasks that share the same task structure but involve entirely new stimuli. Importantly, no explicit meta-learning objective is used. During this task schema learning phase, the RNN exhibits learning-to-learn behavior: the number of trials required to reach the criterion decreases significantly across problems (Fig.2b). An exponential fit reveals a clear trend of accelerated learning, suggesting the RNN utilizes prior knowledge for more efficient adaptation. This pattern mirrors learning-to-learn behavior observed in rats performing the odor-sequence task.

A hallmark of schema learning observed in rats' OFC is the emergence of structured neural representations [4], as shown in Fig.2c. To assess this in our model, we replicate the experimental analysis by computing pairwise distances between trial types in the RNN's hidden state space across positions and sequences, yielding an 24×24 RDM (see Appendix Sec.C.1 for details). This RDM reflects the

structured geometry of the task representation space. As shown in Fig.2c,d, the RNN's RDM closely aligns with that observed in the OFC. Quantitatively, the Pearson correlation between the model and the experimental RDM exceeds 90% (Fig.2e), indicating that the RNN captures the hierarchical task representation structure and exhibits a representational geometry similar to OFC (see Fig.S2). To test the role of shaping, we conduct an additional control experiment in which the RNN is trained directly on a set of odor-sequence problems without shaping. This control model exhibits markedly lower correlation with OFC ensembles than shaped RNNs, indicating that shaping is important for the emergence of structured task representation in the RNN.

4.2 Low-Dimensional Sequence Attractors Serve as Schema for Generalization

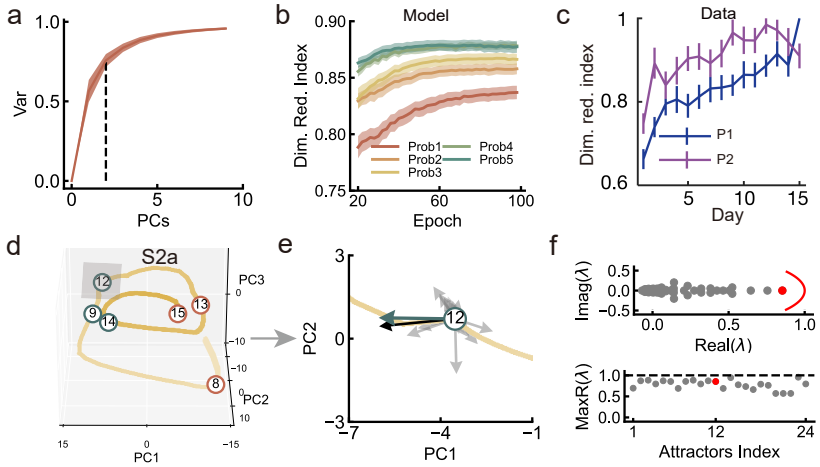

Figure 3: Dimensionality and Attractor Dynamics in a Shaped RNN. (a) Cumulative variance of RNN activity explained by principal components (PCs). Dashed line marks the top three PCs.(b) Dimensionality compression over time in the RNN across five odor-sequence problems. Dim.red.index, dimensionality reduction index - is the normalized variance explained by the top three PCs. Shaded areas indicate the standard error of the mean averaged across 20 seeds. (c) Dimensionality compression over time in rat OFC across two odor-sequence problems [4].(d) PCA visualization of neural trajectories for four sequences (S1a, S1b, S2a, S2b) in the shaped RNN, projected onto the top three principal components. Six fixed points per sequence were identified via optimization-based analysis. (e) Zoom-in of the shaded area in (d). Black arrow: empirical transition direction at the odor 12 fixed point. Green arrow: principal eigenvector; gray arrows: other eigenvectors of the Jacobian matrix at odor 12. (f) Top: Eigenvalue spectrum of the Jacobian at the odor 12 fixed point. Bottom: Maximum eigenvalues across all 24 fixed points, with the red dot indicating the one corresponding to odor 12. See Appendix Sec.A for detailed parameters.

Our brains can reduce the dimensionality of neural representations during schema formation by averaging unique problem details while preserving cross-problem commonalities [3]. To examine this in our RNN, we perform PCA on hidden activities during task schema learning. Fig. 3a shows the network dynamics reside in a low-dimensional subspace, with the top three PCs explaining nearly 80% of the variance. We further track dimensionality change over learning. Fig. 3b shows that the explained variance of the top 3 PCs progressively increases within and across problems, indicating dimensionality compression of neural activity over learning, similar to OFC activity in rats (Fig. 3c).

Together, these findings suggest that a low-dimensional schema code emerges in RNNs trained via shaping, evident in both behavior and representation. However, the underlying neural dynamics of this code remain unclear. To investigate this, we visualize the RNN's state dynamics using PCA. As shown in Fig.3d and Fig.S3, the four sequences (S1a, S1b, S2a, and S2b) form distinct low-dimensional trajectories in the RNN state space. Using optimization-based fixed point analysis [15], we identify six fixed points along each trajectory, each fixed point corresponding to a specific odor's position within the sequence (detail methods see Appendix Sec.C.2). Sequences sharing similar reward and transition structures (e.g., S1a, S1b, S2a) have close trajectories, indicating abstraction of representations with structural commonalities. To probe the stability of these fixed points, we

examine the eigenvalues of the local Jacobian matrix (Fig. 3f). All maximal eigenvalues are below 1, indicating that the fixed points are locally stable and act as attractors. Notably, the principal eigenvector - associated with the largest eigenvalue - aligns with the direction of sequential transitions (Fig. 3e), suggesting it mediates movement along the trajectory.

These results reveal that the RNN encodes odor sequences as sequence attractors - sequences of stable fixed points connected by low-dimensional manifolds. Through a perturbation study (Fig.S4), we further demonstrate that this sequence attractor schema serves as a reusable flexible template, enabling new odor cues to bind and thereby facilitating efficient generalization in novel situations.

4.3 Gradual Emergence of Sequence Attractor Schema Through Shaping

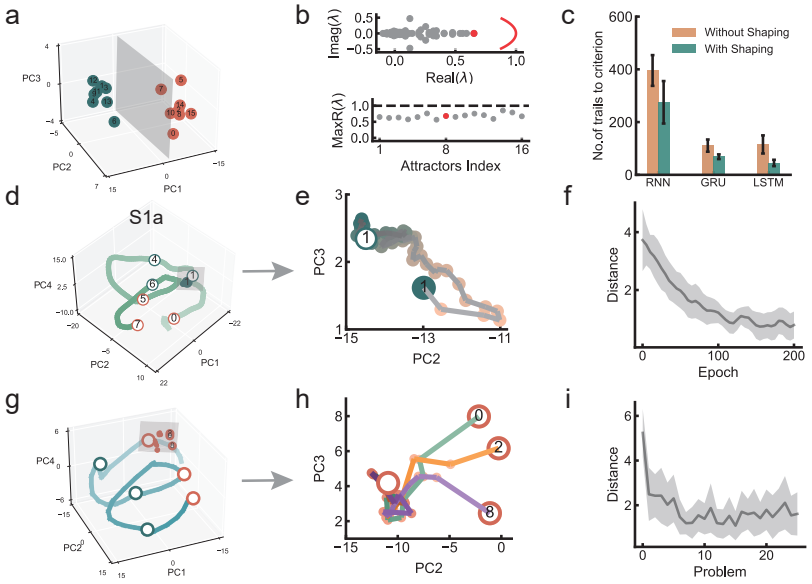

Figure 4: Evolution of attractor dynamics in RNNs during shaping. (a - c) Task primitive learning. (a) PCA visualization of identified fixed points; red = rewarded states, blue = non-rewarded. (b) Stability analysis of fixed points: top, eigenvalue spectrum of a sample odor 8; bottom, maximal eigenvalues across fixed points. (c) Learning efficiency in the subsequent sequence learning stage, comparing models with and without prior task primitive learning, averaged over 5 seeds. Note that the total trial number to criterion includes both task primitive and sequence learning stages for a fair comparison. (d - f) Task sequence learning. (d) PCA visualization of attractor dynamics reveals the gradual shift of the discrete attractor for odor 1 (primitive learning) toward its position in the sequence attractor associated with S1a. Green curve: neural trajectory of S1a. Solid circle: prior attractor from task primitive learning; hollow circles: attractors of S1a sequence. (e) Zoom-in on the learning trajectory of the odor 1 attractor. (f) The average Euclidean distance between evolving attractor states and their corresponding locations of sequence attractors (task sequence learning) over training epochs (averaged over 5 seeds). (g - i) Task schema learning. (g) PCA visualization of attractor abstraction. Green curve: converged abstract neural trajectory of S1a, S1b, and S2a. Numbered hollow circles: prior attractors (sequence learning). Unnumbered hollow circles: attractors of the abstract trajectory (task schema learning). (h) Zoom-in of the abstraction process in (g), illustrating the migration of prior attractors: odor 0 at S1a, odor 2 at S1b, and odor 8 at S2a (sequence learning), toward the abstract attractor within the final sequence attractors (schema learning). (i) The average Euclidean distance between attractors of corresponding sequence steps in S1a, S1b, and S2a, tracked during problem training (averaged over 5 seeds). See Appendix Sec.A for detailed parameters.

Given that sequence attractors serve as a schema, we next investigate their evolution during shaping. We analyze attractor dynamics across three shaping stages. First, during task primitive learning (Fig. 4a, b), the RNN learns simple odor-reward associations and develops 16 discrete, locally stable attractors, each representing a specific odor cue. Importantly, RNNs initialized with these prior discrete attractors learn the subsequent task sequence significantly faster, requiring fewer trials to

reach criterion (Fig. 4c). This learning benefit is also consistent across diverse recurrent architectures, such as LSTM and GRU.

During task sequence learning, we track how prior attractor states in the RNN's state space evolve and examine whether they're reused to form sequence attractors, as shown in Fig. 1b. At each training step, we probe the RNN's attractor states. Taking odor 1 as an example, we initialize the hidden state to its prior attractor and iterate the network dynamics without input until convergence to a new stable state. This new state is then used as the initial state for the next training batch iteration, repeating this process to generate the evolving trajectory of stable states. Fig. 4d,e illustrate the gradual migration of odor 1's prior attractor (primitive learning) to its corresponding location within the S1a sequence attractor structure (sequence learning). We found that 87% (14 out of 16) of prior attractors successfully evolve to their correct target locations. Quantitatively, the average distance between these successfully evolving states and their corresponding attractor locations (sequence learning) steadily decreases over training (Fig. 4f, detail methods see Appendix Sec.C.3). These findings suggest that attractors learned during task primitive learning are reused, reorganized, and linked to form sequence attractors during sequence learning.

During task schema learning, we examine how sequence-specific attractors are compressed into abstract sequence attractors. Figures 4g,h illustrate this process: the attractors for odor 0 in S1a, odor 2 in S1b, and odor 8 in S2a gradually converge toward a common attractor. The average distance between attractors occupying the same sequence position across S1a, S1b, and S2a also decreases over training (Fig. 4i). These results suggest that as the network learns related problems presented consecutively, it gradually averages out sequence-specific details while retaining commonalities shared across S1a, S1b and S2a - eventually forming a unified abstract sequence attractor. This convergence is expected, given the common reward transition structure across these sequences.

Together, these findings reveal a dynamic, attractor-based mechanism for schema formation through shaping. First, discrete attractors emerge during task primitive learning. These are then organized into sequence-specific structures during task sequence learning. Finally, in task schema learning, these structures are compressed and abstracted into a unified schema. This sequence attractor-based schema shaping process supports our conceptual idea shown in Fig. 1b.

4.4 Divergent Learning Dynamics in RNNs with and without Shaping

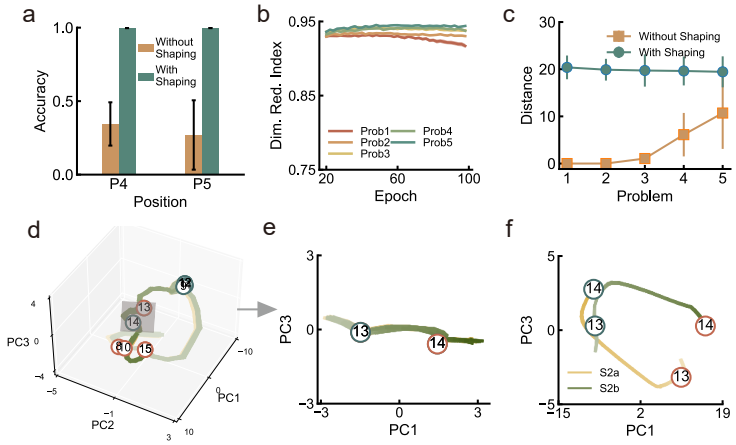

Figure 5: Evolution of attractor dynamics in RNNs without shaping. (a) Performance comparison of RNNs with and without shaping at P4 and P5 in S2a/S2b. Networks were trained and evaluated after problem 1 learning (averaged over 5 seeds). (b) Dimensionality compression dynamics in RNNs without shaping across five odor-sequence problems (averaged over 5 seeds). Other settings are identical to Fig. 3b. (c) Average Euclidean distance between the evolving attractor states corresponding to P4 and P5 across S2a and S2b sequences for RNNs with and without shaping during problem training (averaged over 5 seeds). (d) PCA visualization of sequence attractors in a representative RNN without shaping. Sequence attractors in S2a and S2b converge into a single trajectory after problem 1 learning. (e) Zoomed-in view of attractor states for P4 and P5 in S2a and S2b. (f) PCA visualization of attractor states at the same corresponding positions in problem 5 as shown in (e).

To elucidate the mechanisms underlying behavioral differences between shaped and unshaped RNNs, we examine the learning dynamics of unshaped networks.Shaped RNNs achieve markedly higher reward prediction accuracy ($99.89 \pm 0.03\%$) than unshaped ones ($93.20 \pm 2.4\%$), with the largest discrepancy in classifying odors at P4 and P5 of S2a/S2b, where reward contingencies are reversed. As shown in Fig. 5a, shaped RNNs reach near-perfect accuracy, whereas unshaped RNNs reach only 25% after problem 1. Applying the same PCA-based dimensionality analysis as in Fig. 3b (Fig. 5b), we find that the dimensionality reduction in unshaped RNNs differ qualitatively: (1) Premature compression – unshaped networks enter a low-dimensional regime (93%) much earlier than shaped ones (82.5%), indicating an early collapse to coarse representations; and (2) Lack of progressive abstraction – unlike shaped networks, unshaped ones show little change in dimensionality across problems. These findings reveal fundamentally distinct learning dynamics between shaped and unshaped RNNs.

We further analyze and visualize sequence attractors to better understand representational evolution. In shaped networks, the task-primitive learning stage already establishes four distinct attractors for the odor events at P4 and P5 (in S2a and S2b). These attractors serve as stable building blocks for higher-level, sequence-specific attractor trajectories in later stages, maintaining a consistent distance between corresponding attractor states (Fig. 5c). In contrast, unshaped networks lack this scaffold. Early in training, they represent P4 and P5 odors in S2a and S2b with overlapping trajectories, and solving the task therefore requires the later formation of separate attractor trajectories - a process that is often unstable and leads to failed convergence. Fig. 5d–f shows a successful example of late attractor trajectory separation, where the trajectory for problem 5 diverges from the shared trajectory established after learning problem 1.

In summary, shaped networks build schemas progressively using scaffolded attractor dynamics and structured compression, whereas unshaped networks exhibit abrupt, shallow compression that leads to poor generalization. These findings demonstrate that shaping fundamentally alters the learning trajectory and attractor organization of RNNs, clarifying how and why shaping facilitates learning.

4.5 Sequence Attractor-Based Shaping Improves Learning Efficiency in Keyword Spotting

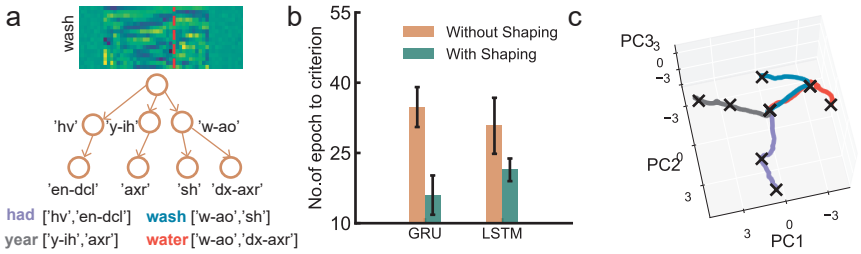

Figure 6: Sequence attractor-based shaping in a keyword spotting task. (a) The task involves four keywords: "water," "wash," "had," and "year." The upper panel shows the MFCC representation of a sample utterance of "water," while the lower panel illustrates the tree-structured phoneme organization of keywords. (b) Learning efficiency comparison (epochs to 90% accuracy) between models trained with and without shaping, averaged over 5 seeds. (c) PCA visualization of RNN neural trajectories after shaping. Crosses indicate identified fixed points. See Appendix Sec.D.2 for detailed parameters.

Based on the sequence attractor-based shaping framework, we apply our method to a real-world sequence task: keyword spotting (Task details see Appendix Sec.D.1). The task involves four spoken words - "water," "wash," "year," and "had" - each composed of basic phonemes forming a hierarchical structure (Fig. 6a). These words are randomly selected from the TIMIT dataset [19]. Following the shaping approach, we divide the task into three stages. In task primitive learning, the model learns to classify phonemes using only one sample per word, forming discrete phoneme-level attractors. In task sequence learning, the model links these attractors by training on sampled word sequences for word class prediction. Finally, in task schema learning, the model trains on the full dataset to abstract a schema and generalize across word variations.

As shown in Fig.6b, shaping significantly improves learning efficiency compared to a model trained without shaping, requiring fewer epochs to reach 90% test accuracy. PCA visualization of hidden

states reveals a tree-like structure consistent with the phoneme hierarchy, supporting the emergence of a sequence schema (Fig.6c). Additionally, we identify seven leaf attractors corresponding to individual phonemes, and one root attractor representing the initial state prior to phoneme onset. These results demonstrate that the sequence attractor-based shaping framework generalizes to complex sequence learning tasks, offering both interpretability and efficiency.

5 Discussion

Sequence schemas form the core of flexible and generalizable intelligence across animals and humans. For complex tasks, acquiring such schemas through pure trial-and-error is often ineffective. Instead, animals and humans use shaping - breaking down complex tasks into simpler subtasks and gradually assembling the full schema. However, the underlying dynamic mechanism behind this process remains unclear. In this work, we use RNNs to study how schemas are represented and evolve through shaping. Our main findings are fourfold: (1) We systematically replicate key behavioral and neural features of schema learning observed in the OFC of rats. (2) We show that sequence schemas can be encoded as sequence attractors. (3) We identify a novel dynamic process of schema formation, progressing from point attractors to sequence attractors, and finally to abstract schemas. (4) We demonstrate the practical utility of this framework by successfully applying it to keyword spotting. These findings may advance understanding of the neural mechanisms of schema learning via shaping and offer new insights into how abstract, generalizable structures can be learned from experience in both biological and artificial systems.

Related Works. Schema learning has attracted increasing attention in neuroscience [3, 10, 13, 15, 20, 21, 22, 23, 27]. Previous studies have primarily examined schema representations - for example, low-dimensional manifolds or attractors in sensorimotor [13], decision-making tasks [15], or capturing temporal order [20]. However, a fundamental question remains overlooked: how does the brain learn schemas through shaping? Despite its prevalence in animal training, the mechanism of shaping is systematically underexplored in neuroscience. And previous works mainly focus on behavioral outcomes, neglecting the underlying dynamics of abstract knowledge acquisition [21, 22]. Our work addresses this deficit by investigating schema learning via shaping, providing the first explicit link between these two essential concepts.

Our approach mirrors curriculum learning by decomposing complex tasks into simpler subtasks to boost learning efficiency [24, 25]. However, while curriculum learning in machine learning primarily focuses on optimizing performance generalization [25, 26], it typically lacks a mechanistic understanding of how the underlying abstract knowledge is acquired during the process [25]. In contrast, we offer a dynamic framework grounded in biological learning: learning progresses from primitive attractor structures to reusable abstract attractor schemas that facilitate generalization [15]. This framework may provide inspiration for new curriculum learning algorithms that explicitly facilitate the learning and use of abstract knowledge.

Recent studies have questioned the identifiability and robustness of RNNs in neuroscience modeling, showing they can achieve similar behaviors via distinct internal solutions [28]. Our analysis, however, reveals that shaped RNNs converge to highly similar internal dynamics regardless of initialization. Applying Canonical Correlation Analysis (CCA) to models trained with different seeds (Fig.S6), we find strong alignment among the top three canonical components, indicating consistent representation geometry across models. Previous findings show that task complexity or multi-task learning can reduce degeneracy in RNN solutions [28, 29], we extend this by demonstrating that our shaping paradigm may be an effective way to reduce this degeneracy.

Limitations and Future Works. A limitation of this work is that model optimization relies on backpropagation through time, which is not biologically plausible [30]. Future work could explore combining dopamine-based reward learning [31] with slow Hebbian plasticity [32] to develop more biologically grounded mechanisms for schema learning. Another limitation is that our model includes only the PFC, whereas schema formation in the brain is thought to result from dynamic interactions between the PFC and the hippocampus [17, 33, 34], which play complementary roles with distinct learning timescales [35]. The hippocampus may support task primitives and sequence learning, while the PFC contributes to task schema learning. Modeling their interaction could help clarify the neural basis of schema formation through shaping and inspire more flexible and generalizable sequence learning algorithms.

Acknowledgments and Disclosure of Funding

This work was supported by the National Science and Technology Innovation 2030 Major Program (No. 2021ZD0203700 / 2021ZD0203705, Y.Y. Mi), National Natural Science Foundation of China (62336007, B. Hong).

References

[1] Gilboa, A., & Marlatte, H. (2017). Neurobiology of schemas and schema-mediated memory. Trends in cognitive sciences, 21(8), 618-631.

[2] Tse, D., Langston, R. F., Kakeyama, M., Bethus, I., Spooner, P. A., Wood, E. R., ... & Morris, R. G. (2007). Schemas and memory consolidation. Science, 316(5821), 76-82.

[3] Bein, O., & Niv, Y. (2025). Schemas, reinforcement learning and the medial prefrontal cortex. Nature Reviews Neuroscience, 1-17.

[4] Zhou, J., Jia, C., Montesinos-Cartagena, M., Gardner, M. P., Zong, W., & Schoenbaum, G. (2021). Evolving schema representations in orbitofrontal ensembles during learning. Nature, 590(7847), 606-611.

[5] Harlow, H. F. (1949). The formation of learning sets. Psychological review, 56(1), 51.

[6] Ma, F., Lin, H., & Zhou, J. (2025). Prediction, inference, and generalization in orbitofrontal cortex. Current Biology, 35(7), R266-R272.

[7] Maor, I., Atwell, J., Ascher, I., Zhao, Y., Takahashi, Y. K., Hart, E., ... & Schoenbaum, G. (2025). Persistent representation of a prior schema in the orbitofrontal cortex facilitates learning of a conflicting schema. bioRxiv, 2025-02.

[8] Makino, H. (2023). Arithmetic value representation for hierarchical behavior composition. Nature neuroscience, 26(1), 140-149.

[9] Mi, Q., & Summerfield, C. (2025). Human curriculum learning of a cue combination task.

[10] Lee, J. H., Mannelli, S. S., & Saxe, A. (2024). Why do animals need shaping? a theory of task composition and curriculum learning. arXiv preprint arXiv:2402.18361.

[11] Mante, V., Sussillo, D., Shenoy, K. V., & Newsome, W. T. (2013). Context-dependent computation by recurrent dynamics in prefrontal cortex. nature, 503(7474), 78-84.

[12] Inagaki, H. K., Fontolan, L., Romani, S., & Svoboda, K. (2019). Discrete attractor dynamics underlies persistent activity in the frontal cortex. Nature, 566(7743), 212-217.

[13] Goudar, V., Peysakhovich, B., Freedman, D. J., Buffalo, E. A., & Wang, X. J. (2023). Schema formation in a neural population subspace underlies learning-to-learn in flexible sensorimotor problem-solving. Nature Neuroscience, 26(5), 879-890.

[14] Yang, G. R., Joglekar, M. R., Song, H. F., Newsome, W. T., & Wang, X. J. (2019). Task representations in neural networks trained to perform many cognitive tasks. Nature neuroscience, 22(2), 297-306.

[15] Driscoll, L. N., Shenoy, K., & Sussillo, D. (2024). Flexible multitask computation in recurrent networks utilizes shared dynamical motifs. Nature Neuroscience, 27(7), 1349-1363.

[16] Schoenbaum, G., & Eichenbaum, H. (1995). Information coding in the rodent prefrontal cortex. I. Single-neuron activity in orbitofrontal cortex compared with that in pyriform cortex. Journal of neurophysiology, 74(2), 733-750.

[17] Wikenheiser, A. M., & Schoenbaum, G. (2016). Over the river, through the woods: cognitive maps in the hippocampus and orbitofrontal cortex. Nature Reviews Neuroscience, 17(8), 513-523.

[18] Wang, X. J. (2008). Decision making in recurrent neuronal circuits. Neuron, 60(2), 215-234.

[19] Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., Pallett, D. S. & Dahlgren, N. L. (1993). DARPA TIMIT Acoustic Phonetic Continuous Speech Corpus CDROM NIST

[20] Zou, X., Chu, Z., Guo, Q., Cheng, J., Ho, B., Wu, S., & Mi, Y. (2023). Learning and processing the ordinal information of temporal sequences in recurrent neural circuits. Advances in Neural Information Processing Systems, 36, 33999-34020.

[21] Krueger, K. A., & Dayan, P. (2009). Flexible shaping: How learning in small steps helps. Cognition, 110(3), 380-394.

[22] Hocker, D., Constantinople, C. M., & Savin, C. (2024). Compositional pretraining improves computational efficiency and matches animal behavior on complex tasks. bioRxiv.

[23] Bruch, S., McClure, P., Zhou, J., Schoenbaum, G., & Pereira, F. (2021). Validating the Representational Space of Deep Reinforcement Learning Models of Behavior with Neural Data. bioRxiv, 2021-06.

[24] Bengio, Y., Louradour, J., Collobert, R., & Weston, J. (2009). Curriculum learning. In Proceedings of the 26th annual international conference on machine learning (pp. 41-48).

[25] Wang, X., Chen, Y., & Zhu, W. (2021). A survey on curriculum learning. IEEE transactions on pattern analysis and machine intelligence, 44(9), 4555-4576.

[26] Bauer, J., Baumli, K., Behbahani, F., Bhoopchand, A., Bradley-Schmieg, N., Chang, M., ... & Zhang, L. M. (2023). Human-timescale adaptation in an open-ended task space. In International Conference on Machine Learning (pp. 1887-1935). PMLR.

[27] Khona, M., & Fiete, I. R. (2022). Attractor and integrator networks in the brain. Nature Reviews Neuroscience, 23(12), 744-766.

[28] Huang, A., Singh, S. H., Martinelli, F., & Rajan, K. (2025). Measuring and controlling solution degeneracy across task-trained recurrent neural networks. ArXiv, arXiv-2410.

[29] Yang, G. R., Cole, M. W., & Rajan, K. (2019). How to study the neural mechanisms of multiple tasks. Current opinion in behavioral sciences, 29, 134-143.

[30] Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J., & Hinton, G. (2020). Backpropagation and the brain. Nature Reviews Neuroscience, 21(6), 335-346.

[31] Wang, J. X., Kurth-Nelson, Z., Kumaran, D., Tirumala, D., Soyer, H., Leibo, J. Z., ... & Botvinick, M. (2018). Prefrontal cortex as a meta-reinforcement learning system. Nature neuroscience, 21(6), 860-868.

[32] Hattori, R., Hedrick, N. G., Jain, A., Chen, S., You, H., Hattori, M., ... & Komiyama, T. (2023). Meta-reinforcement learning via orbitofrontal cortex. Nature Neuroscience, 26(12), 2182-2191.

[33] Lin, H., & Zhou, J. (2024). Hippocampal and orbitofrontal neurons contribute to complementary aspects of associative structure. Nature Communications, 15(1), 5283.

[34] Zou, X., Cao, X., Yang, X., & Hong, B. (2024). Leveraging attractor dynamics in spatial navigation for better language parsing. In Forty-first International Conference on Machine Learning.

[35] Kumaran, D., Hassabis, D., & McClelland, J. L. (2016). What learning systems do intelligent agents need? Complementary learning systems theory updated. Trends in cognitive sciences, 20(7), 512-534.

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 abstract and introduction clearly state the main claim, the evidence is provided in Section 4.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: We discuss limitations in the section 5.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [NA]

Justification: This paper does not include formal theoretical results or proofs, as it focuses on neural dynamical modeling and empirical analysis without introducing new theorems.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The model architecture, training procedures, and experimental setup are detailed in Section 3 and the technical appendices.

   Guidelines:

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

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

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

Answer: [No]

Justification: Code is not publicly available at submission, but Section 3 and technical appendices provide detailed instructions to reproduce the main results.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Training and test details are fully described in Section 3 and technical appendices.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Error bars representing the standard deviation over multiple runs with different random initializations are reported in Fig.2a,e; Fig.3a; Fig.4c,f,i; and Fig.6b, while the standard error of the mean (SEM) is reported in Fig. 2b.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Experiments ran on an Intel i9-14900k CPU with 32GB RAM, no GPU used. Each training took 30 minutes, with 200 runs for tuning and validation.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: Our research adheres fully to the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [NA]

    Justification: This work is foundational and theoretical, without immediate societal applications or impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [NA]

    Justification: This work does not release any models or datasets that pose high risks of misuse.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We use the TIMIT dataset (LDC93S1) with proper citation, under our institution's license (LDC User Agreement for Non-Members).

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [NA]

    Justification: The paper does not introduce any new assets. It only uses existing available datasets and standard models.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: The research does not involve any crowdsourcing or human subjects.

    Guidelines:

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

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

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

    Answer: [NA]

    Justification: The research does not involve crowdsourcing or human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [NA]

    Justification: LLMs were not involved in the core methods.

    Guidelines:

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