Path Integration and Object-Location Binding Emerge in an Action-Conditioned Predictive Sequence Network

Linda Ariel Ventura[*1, 2], **Victoria Bosch**[*1], **Tim C Kietzmann**[1] **& Sushrut Thorat**[1]
[1]Institute of Cognitive Science, Osnabrück University, Germany
[2]Sorbonne University, France
[*]Shared first authorship

Abstract

Adaptive cognition requires structured internal models of objects and their relations. Predictive neural networks are often proposed to learn such world models, but how these are instantiated and how they support prediction remain unclear. We investigate this in a minimal in-silico setting. A recurrent neural network samples tokens sequentially from 2D continuous token scenes and is trained to predict the upcoming token from the current input and a saccade-like displacement. On novel scenes, prediction accuracy improves across the sequence, indicating in-context learning. Decoding analyses reveal path integration and dynamic binding of token identity to position. Interventional analyses show that new bindings can be learned late in sequence and that out-of-distribution bindings can be learned as well. Together, these findings show how structured representations relying on flexible binding emerge to support prediction, offering a mechanistic account of sequential world modeling relevant to cognitive science.[1]

Keywords: in-context learning; integration; binding; prediction; memory

Introduction

Understanding and operating in the natural world requires representing objects and their relations, and updating those representations as new information is acquired. How could biological agents acquire such structured models of the world? A notable proposal in cognitive science and machine learning is that such structured internal world models can emerge from learning to predict future sensory inputs, notably when prediction is conditioned on the agent's own actions (i.e., efference copies; see Figure 1A; Brown et al., 2020; Elman, 1990; Ha and Schmidhuber, 2018; LeCun, 2022; O'regan and Noë, 2001; Petroni et al., 2019). For example, cognitive science accounts of sensorimotor contingencies and affordance-based cognition emphasize that perception is structured by the expected sensory consequences of possible actions (Eperon et al., 2026; O'regan & Noë, 2001). While both biological agents and predictive neural networks appear capable of acquiring structured models, it remains unclear how these "world models" are implemented internally: what mechanisms encode the sensed parts of the world and their relationships, and retrieve the relevant parts for prediction?

One recent example of a system performing action-conditioned prediction is the Glimpse Prediction Network (GPN), where predicting the content of the next fixation in a sequence of eye movements, conditioned on a saccadic efference copy, drives integration across time and yields unified scene representations aligned with human neural responses to natural scenes (Thorat et al., 2025). To probe the mechanisms supporting such model-based sequence prediction, we study a minimal setting inspired by GPN. Our goal is to retain the core ingredients of such action-conditioned prediction, while

[1]The code to reproduce these results can be found at: https://github.com/KietzmannLab/minimal_world_model_interp

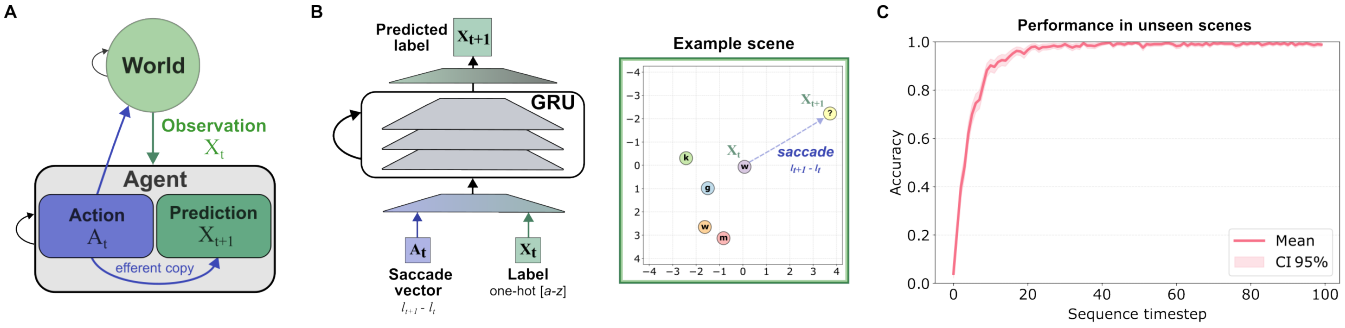

Figure 1: **A minimal action-conditioned predictive sequence network. A.** Conceptual figure depicting action-conditioned prediction. **B.** Our instantiation of a prediction network, and an example of a scene with 6 tokens placed in a 2D continuous space. The network receives the saccade displacement vector and the label of the current token, and is tasked to output the label of the subsequent token in the sequence. **C.** Average network prediction performance in unseen worlds (N=500) per sequence timestep. The network quickly learns the structure of novel worlds, using its dynamics without any parameter changes (in-context learning). Error envelope depicts 95% CIs of the means.

In G. Boleda, I. Dautriche, M. Maldonado, C. Saldana and T. Torrent (Eds.), *Proceedings of the 48th Annual Conference of the Cognitive Science Society*

stripping away domain-specific complexity (e.g., visual co-occurrence and semantics of scene parts). This allows for increased interpretability and ease of probing the network's internal mechanisms. Here, we consider scenes as sets of tokens in a two-dimensional continuous space, and construct sequences of displacements between tokens (saccades). At each step, a recurrent neural network (RNN) predicts the label of the next token given the current token and the provided displacement.

Training to predict the next token in sequences over many scenes induces an in-context learning ability to encode tokens and their relative positions in novel scenes, without any weight updates. Because the latent structure of the scenes is fully known, this setting allows precise interrogation of the network's internal representations of the scene components and their relations. To identify mechanisms and decompose computational components underlying this behavior, we formulate a hypothesis space using a symbolic algorithm for saccade-conditioned token encoding and retrieval. We show that the hypothesized algorithmic components exist in the network: path integration of saccades and binding of tokens to absolute positions. Interventional analyses demonstrate that the network can memorize new label-position bindings in-context while retaining previously stored bindings, and the binding operation extends to out-of-distribution token-position pairs. Together, these results demonstrate how an action-conditioned prediction objective can give rise to mechanisms that implement a structured model of the observed world. More broadly, this work demonstrates how mechanistic interpretability analyses of a minimal network can reveal algorithmic components that might underlie world modeling in more complex predictive systems.

Methods

Minimal scene construction

As seen in Figure 1B, each minimal scene consists of $4-6$ tokens sampled from the 26 possible letters of the English alphabet (a-z); letters can occur multiple times. These tokens are placed on a continuous 2D space with x/y coordinate bounds of $[-4, 4]$ and a minimal distance of 0.25 between tokens. As a result, the space of possible label-position combinations is massive, allowing for on-the-fly data generation during training without substantial overlap between examples. Saccade sequences are sampled from these scenes: at any given timestep, a random displacement to one of the other tokens is initiated. The first timestep always corresponds to the token at the center of the scene.

Network architecture & training

As seen in Figure 1B, the network that we tasked with this simplified next-token prediction is a 3-layer Gated-Recurrent Unit (GRU) RNN (Cho et al., 2014; Paszke et al., 2019), with a hidden state size of 512. The 26D one-hot token label input and 2D saccade (displacement between two tokens) input are linearly projected to a 512D layer, which is fed as input to the GRU. The output of the GRU is projected to a 512D ReLU layer, from which a 26D linear readout predicts the upcoming token label at the next timestep. We train the network until convergence (for 40960 batches of 200 scenes each; one sequence per scene; sequence length of 100 timesteps), using cross-entropy loss. As any token can be present at any position in the scenes, we expect the network to rely on its internal dynamics to learn about the token distributions in the scene being sensed through the input sequence.

Robustness Results reported in this paper were qualitatively reproduced with two more different random seed initializations, as well as in a 3-layer parameter-matched LSTM. All statistics reported in this manuscript are acquired using two-sided t-tests.

Results

In-context learning of token arrangement in scenes

After training the recurrent neural network to predict the next token in sequences over many scenes, we probe its generalization capacity by testing the trained frozen network on sequences in newly-generated scenes. The network predicts the upcoming token with increasing accuracy as the sequence proceeds (Figure 1C; N = 500 scenes). The network thus demonstrates capacity for in-context learning of the token arrangement in scenes (Olsson et al., 2022), and reaches peak prediction accuracy within 35 timesteps.

Next, we ask how the network is capable of learning a "world model", i.e., the tokens and their relative positions in scenes, and what format and mechanisms it uses to store this knowledge for the retrieval of the next token without changing its weights.

Algorithm 1 Hypothesized Symbolic Algorithm

Initialize dictionary $D \leftarrow \emptyset$ & position $p \leftarrow (0, 0)$
for $t = 1$ to T **do**
    Observe token label X_t and intended saccade A_t
    Store $D[p] \leftarrow X_t$
    Update position: $p \leftarrow p + A_t$
    if p is a key in D **then**
        Output stored label $D[p]$
    else
        Output a random label from alphabet $\mathcal{A}$
    end if
end for

Signatures of path integration and token label-position binding

As a starting point, we hypothesize a symbolic algorithm that the network could implement (Algorithm 1). The network must memorize observed tokens and process their relative positions. A general memory-efficient solution is a dictionary-like format that binds positions to labels, i.e. *{position: label}*.

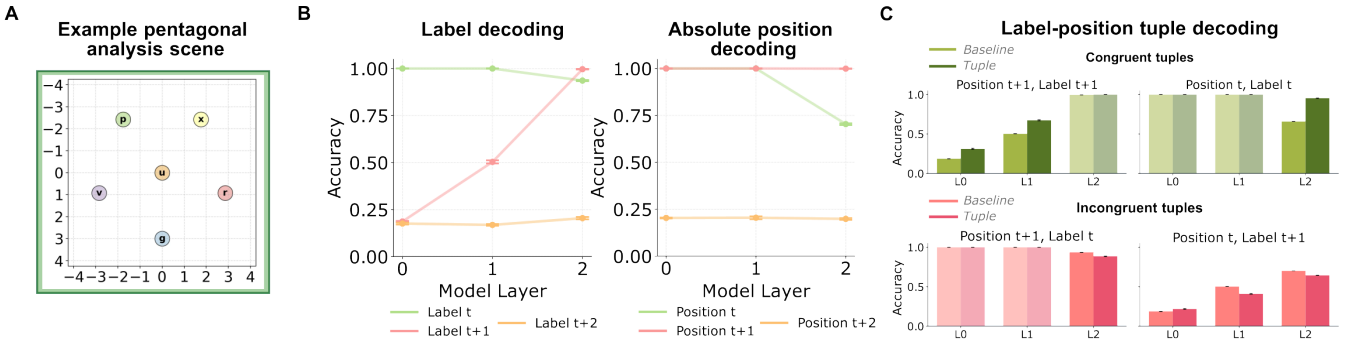

Figure 2: **Assessing algorithmic components in the network's representations. A.** Example pentagonal scene used for the decoding analyses. We generate 500 test scenes with this arrangement. **B.** Decoding of token label and absolute token position across network layers, averaged across timesteps 35 – 100. High absolute position decoding suggests the network learned path-integration. **C.** Decoding of label-position tuples across network layers, averaged across timesteps 35 – 100, compared to their corresponding baselines, i.e., the expected decoding accuracy based on the decoding of the components (product of their decoding accuracies). The top panels show results for congruent label-position pairs, and the bottom panels depict results for incongruent label-position pairs. We highlight the results for which the baseline accuracy is below 1, as in the cases of perfect component decoding we cannot meaningfully measure accuracy deviations that would indicate binding. The congruent tuples, but not the incongruent tuples, can be decoded better than their corresponding baselines, suggesting that the positions of tokens were bound to their labels resulting in a distinct representation compared to the parts. Error bars depict 95% CIs of the means.

Compared with a transition cache of local tuples, *(label1, saccade, label2)*, this format requires $O(N)$ rather than $O(N^2)$ memory, where N is the number of tokens. Crucially, a bound label-position format allows post-hoc inference of unseen token relations, as required when any token can be queried by a saccade. Consistent with this, the network zero-shot infers the true next token for saccades withheld for the first 100 timesteps (99.2% accuracy; N = 500 scenes), suggesting that it encodes absolute token positions. This favors position-based binding over a transition cache, which would not support such inference.

To construct a bound label-position representation, we hypothesize two capabilities are necessary: 1) the network needs to path integrate sequentially to infer the absolute position at a given timestep, and 2) bind the currently-seen token to the inferred absolute position and store it. A final component for the prediction is retrieval: at a timestep, given the saccade and performing path integration, the network has to be able to retrieve the label of the next token at the inferred position, and output the predicted token. To search for the components of such an algorithm, we set up a controlled scene, sampling 6 tokens and arranging them as a pentagon (radius of the circumscribing circle = 3) and its center (see Figure 2A; 500 scenes generated for testing the network). We commence by asking whether we can decode the token labels and positions at different timesteps from the layer activations of the network (across sequence timesteps 35 – 100), using Support Vector Machine (SVM) classifiers with 5-fold cross-validation (Cortes & Vapnik, 1995; Pedregosa et al., 2011). We decode token label (26-way classification) and absolute position (6-way classification) at the current timestep (t), as well as the two subsequent timesteps ($t + 1$ and $t + 2$). Here, $t + 2$ acts as a baseline, as neither token label nor position for this timestep can be known from the history, currently available, or predicted information. Indeed, we observe chance performance (20%) throughout the layers for both label and position decoding at timestep $t + 2$.

Given the hypothesized processes related to binding and retrieval, we expect the current and predicted token identities to be represented in the network layers. Indeed, as seen in Figure 2B (left), we observe high label decoding for the current token in all layers, whereas the accuracy for the next token increases with layers throughout the network. This result is intuitive, because the current token is provided as input, whereas the next token has to be inferred and serves as the output of the model.

The path integration hypothesis predicts both the current and predicted token positions to be represented in the network layers. We observe perfect absolute position decoding for both timesteps in the first layer, and a small decrease in accuracy with layer depth, more so for the current position than the predicted position (Figure 2B; right). High linear separability of both the current and subsequent absolute positions signals path integration, as the network only receives relative token positions (saccades).

Having confirmed that the components (token label and position) are present in the network representations, we move on to search for evidence of them being bound together. We test whether the network encodes bound representations of label and absolute position, i.e., whether congruent label-position tuples are more decodable ($26 \times 6 = 156$-way classification) than would be expected from a joint-decoding baseline, which is the product of the decoding accuracies for the token label and the position separately. Note that if the components are perfectly decodable, i.e, baseline accuracy = 1, we cannot

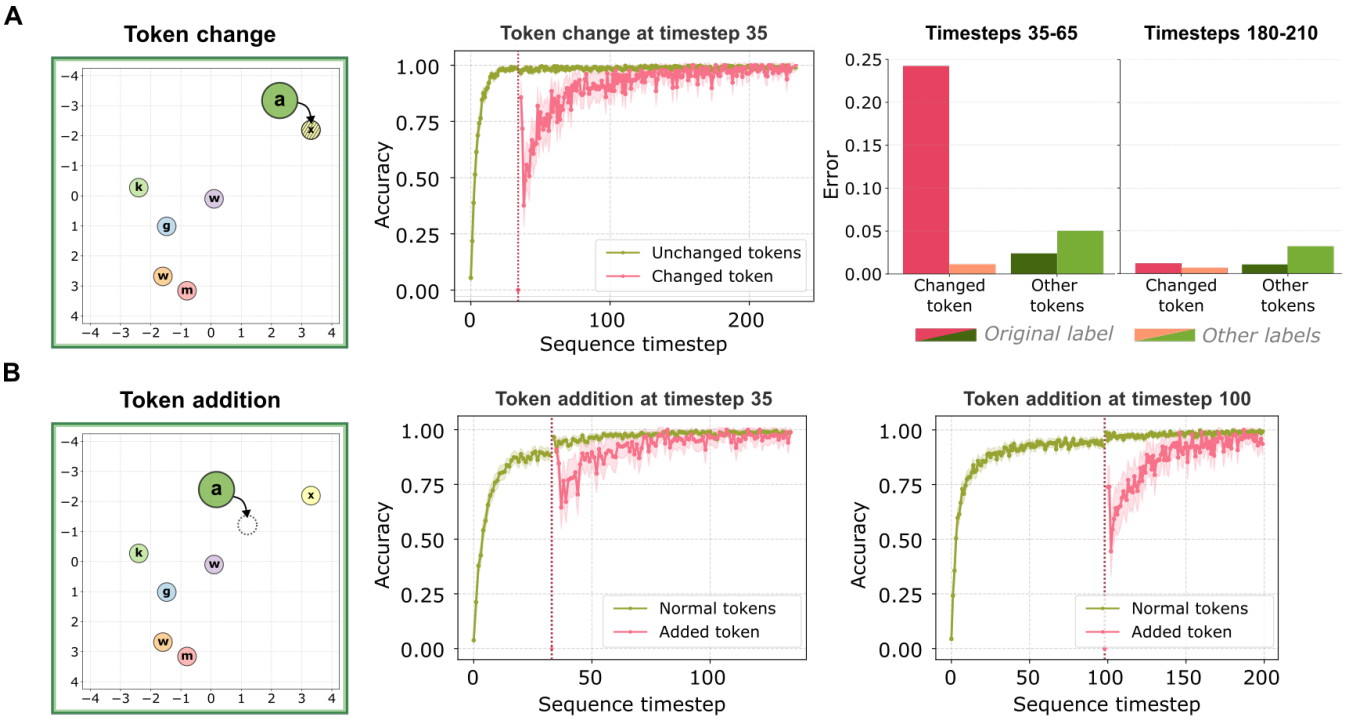

Figure 3: **Assessing the stability and plasticity of in-context memory. A.** Left: Illustration of the token change procedure. Middle: Performance under intervention for tokens at the changed and unchanged locations (N= 500 scenes). Right: Error contribution of original replaced labels and other labels at other locations. When a token at a particular location is changed, the network gradually switches to predicting the new token at that location, suggesting short-timescale stability of its memory. **B.** Left: Illustration of the token addition procedure. Middle: network performance after token addition at timestep 35 (N= 500 scenes). Right: network performance after token addition at timestep 100 (N= 500 scenes). When a new token is added at a new location, the network requires multiple repetitions of exposure to encode it in memory. It can do so even after prior heavy exposure to the rest of the scene, suggesting a consistently plastic memory. Error envelopes depict 95% CIs of the means.

meaningfully determine whether these components are bound, as the expected tuple decoding will also be 1. As shown in Figure 2C (top panels), decoding accuracy for congruent tuples exceeds their baselines: for the tuple at the upcoming position (timestep $t+1$), decoding accuracy exceeds the baseline at the first two layers (both $p < 0.001$). For the tuple at the current position (timestep t), decoding accuracy is above baseline in layer 2 ($p < 0.001$).

To rule out the possibility that *any* combination of label and position is jointly decodable above baseline, we run a mismatch control. Specifically, we consider cross-timestep tuples by combining the position (or label) at the current timestep with the label (or position) from the subsequent timestep, decoding these incongruent combinations, and comparing them with their corresponding product baselines. Unlike the congruent tuples, these incongruent tuples show little to no elevation, or a reduction compared to their baselines (Figure 2C; all layers differ significantly from baseline, $p < 0.001$). Thus, the joint-decoding boost is selective for the expected (congruent) label-position pairing, consistent with a specific position-label bound representation rather than a generic mixture of decodable components. This provides evidence towards the hypothesized mechanism of binding of label and position in the network.

We reproduce these results in a parameter-matched LSTM where joint decoding of the label-position tuple is better than baseline for congruent tuples (position $t+1$: $p < 0.001$ for all layers; position t: $p < 0.001$ for L2) and baseline decoding improves over incongruent tuples (position $t+1$: $p < 0.001$ for L2; position t: $p < 0.001$ for L0, L1). This indicates that it is plausible that other architectures may solve the task using similar algorithmic components.

The stability, plasticity, and generalizability of in-context scene memory

To further characterize the network's in-context learning ability and our finding that its memory contains bound representations, we use interventional analyses to ask at what stage in the sequence these representations can be introduced or modified, and whether out-of-distribution label-position bindings can be learned.

First, we ask whether we can replace a token after it has been memorized at a position. As we observe convergence of prediction accuracy around timestep 35, we chose to re-

place one of the scene's tokens at that time point and continue the sequence accordingly (see Figure 3A, left panel; N = 500 scenes with 6 tokens each). Measuring the prediction accuracy of the network for the tokens at both the unchanged positions as well as the (new) token at the changed position, we observe that the performance at other positions does not change after the intervention, indicating no observable disruption of the memory representations (Figure 3A, middle panel). For the changed position, we do observe a gradual increase in prediction performance of the new token across sequence steps. To better understand the memory encoding and process underlying these observations, we further investigate the causes of error on the changed position: is the original label-position tuple *overwritten*, or does it keep competing with the newly-introduced tuple? Separating the cause of error by token label, we observe that initially (between timesteps $35 - 65$), the largest share of the errors made on the position of intervention are due to the network erroneously predicting the original label (see Figure 3A, right panel). However, after 180 time steps, this error type decreases significantly in occurrence ($p < 0.001$), revealing that although challenging, increased exposure of the new token at this position does overwrite the original memory of the label-position tuple.

Next, we ask whether we can add in new tokens after the network has converged on its performance on a given test scene. We do so by introducing a novel token at a later timestep to 500 scenes with 5 randomly-picked tokens each (Figure 3B, left panel), and evaluating performance thereafter (while randomly cycling through the 6 tokens). After introducing the new token at timestep 35, the network quickly learns to start predicting the new token at its position, reaching peak accuracy after $\sim$ 50 steps (Figure 3B, middle panel). Does the network's memory become less plastic over time? We observe that, when introducing the new token at timestep 100, the network is still able to learn the new token and predict it with high accuracy after $\sim$ 50 steps as well (Figure 3B, right panel), without changing its weights, indicating that in-context memory plasticity does not reduce over time.

Finally, as a test of the generalizability of in-context memory to unseen token arrangements, we ask if the network can learn to associate a token with positions where it was never seen during training. Specifically, we set up a control setting: during network training the label k is only ever shown in the lower-right quadrant at the control position $(1, 1)$, while no other label is shown at that position or in a 0.25 radius area around it (see Figure 4A). After training, we construct test scenes of 6 tokens, in which a k token is only shown in the other three quadrants of the scene (the quadrants not containing position $(1, 1)$), and one of the other tokens is shown at the control position. We evaluate performance for predicting k, and the other token at $(1, 1)$. We observe that the network can learn to infer k in the three other quadrants, and can infer any other token at the control position (N = 500 scenes; see Figure 4B). This suggests that the network can arbitrarily bind known tokens to known positions, even if during training those

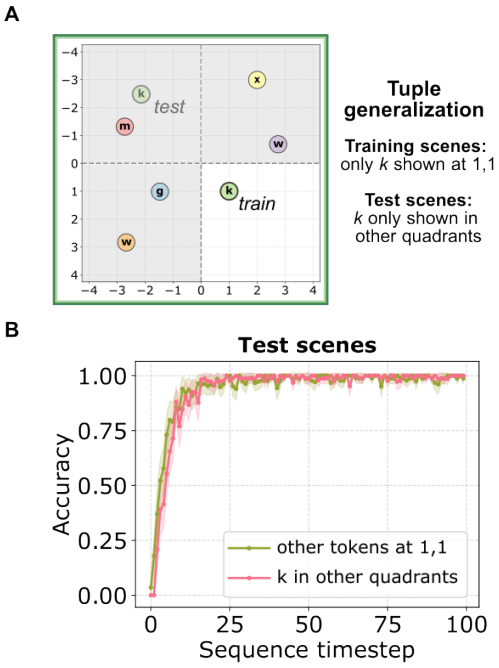

Figure 4: **Assessing the generalizability of token identity-location binding. A.** Illustration of tuple generalization analysis: during training, the token k is only shown at position $(1, 1)$ while other tokens can be placed anywhere on the grid. **B.** Network performance on test scenes with a non-k token placed at the control location $(1, 1)$, and k placed in the other three quadrants (N= 500 scenes). These results suggest that the network can flexibly bind tokens to locations, overriding the label-location co-occurrence learned during training. Error envelope depicts 95% CIs of the means.

tokens were never seen in those positions.

Taken together, these interventional analyses illustrate the stability, flexibility, and generalizability of in-context learning in the network. The network is capable of memorizing new tokens at any point in time, and does not show critical phases in its learning process. Yet, it appears to be challenging for the network to overwrite an existing bound representation, signaling stable memory representations. This suggests that the memory representations are more complex than a simple dictionary-like representation. If it were to be dictionary-like, label changes would happen instantaneously, as once the position is indexed, the associated token could be replaced. Finally, the binding operation is not limited to known label-position distributions but can be extended to new, out-of-distribution label-position pairs.

Discussion

Training a recurrent neural network to predict the next token label from two ingredients (the current token and a saccade displacement) induces a robust in-context learning ability: on newly generated scenes, the frozen network improves next-token prediction as the sequence unfolds, without any weight

updates (Brown et al., 2020; Olsson et al., 2022). Because scenes are generated on the fly in continuous space, this improvement cannot be explained by memorizing a finite set of scenes or fixed label-position patterns. Instead, the network must use the sequence itself to infer the tokens present in the current scene and how they are arranged, and use that inferred structure to answer arbitrary next-token queries driven by the saccade input.

Mechanistically, our analyses point to two core ingredients. First, the network represents absolute position despite receiving only relative displacements, consistent with internal integration of saccades across time. Related computational pressures are known to elicit path integration codes in recurrent networks trained on self-motion and localization tasks (Banino et al., 2018; Cueva & Wei, 2018; Sorscher et al., 2023). Second, beyond representing labels and positions separately, the network shows evidence consistent with the binding of labels and their absolute positions: congruent label-position tuples are selectively more decodable than expected from the components alone, and mismatched controls show that this boost is specific to the correct pairing. Together with correct prediction under saccades not encountered early in the sequence, these findings support the idea that the network forms an internal record akin to "this label is at this position" that can be queried for prediction.

Note that *binding* here refers to a retrieval-oriented label-position association rather than classical feature conjunction. In cognitive science, binding often refers to combining separable perceptual attributes into coherent objects (Kahneman et al., 1992; Scholte & de Haan, 2025; Treisman & Gelade, 1980). Our task instead demands a role-filler style association: a token identity must be linked to a position so that it can be retrieved when a saccade specifies a target. This requirement speaks to longstanding questions about whether distributed connectionist representations support systematic variable binding (Fodor & Pylyshyn, 1988), and connects to work on dynamic binding in structured representations, including synchrony-based proposals and distributed binding schemes (Griffiths et al., 2025; Hummel & Biederman, 1992; Kanerva, 2009; Kleyko et al., 2022; Plate, 1995; Smolensky, 1990). Future work could assess the *format* of the learned association, for example by testing whether network states are well-approximated by superposed role-filler bindings as in tensor product representations, or by treating alternative vector-symbolic binding operators as competing hypotheses in fits to internal activations (Greff et al., 2020; McCoy et al., 2018; Plate, 1995).

We note that, a priori, path integration and binding are not the only possible mechanisms that could solve the sequential prediction task. However, our results challenge alternative mechanisms such as a within-episode transition cache that stores local tuples of the form *(label1, saccade, label2)*. Such a cache would be expected to fail when queried with unseen saccades, whereas we observe robust zero-shot inference under withheld displacements. Moreover, the results of the out-of-distribution manipulation are better explained by a compositional binding mechanism, because it directly tests whether binding depends on label-specific training "support" (e.g., a label being bindable only at positions encountered during training). The network's success indicates that the association operation can generalize to new positions for a label even when its training exposure was spatially restricted. These results reinforce our hypothesis that the mechanisms underlying the network's capacities rely on structured and relational mechanisms, rather than simpler lookup-based strategies.

Interventional analyses further demonstrated that the in-context scene memory encoded by the network is both plastic and stable. New label-position pairings can be acquired late in a sequence, yet overwriting an existing pairing is slow and initially dominated by the original label. This overwrite difficulty argues against a simple dictionary-like memory format and suggests the persistence of old associations that interfere with new ones, echoing classic constraints in connectionist accounts of learning and memory (McClelland et al., 1995; McCloskey & Cohen, 1989).

The mechanisms identified here also connect to proposed hippocampal-entorhinal mechanisms for spatial and relational memory. Path integration and flexible item-location binding are central to accounts of cognitive maps and are explicit in models such as the Tolman-Eichenbaum Machine, where relational structure is learned from sequential experience and generalized to new queries (Whittington et al., 2020). This raises the possibility that related algorithmic components support biological sequence learning. Testing this would require a comparable task in which subjects learn a novel layout through sequential observations and action-related signals, while probing neural activity recordings for integrated position, upcoming position, item identity, and item-position pairings. Such a comparison is nontrivial: efference copies help predict the sensory consequences of action (Wolpert & Flanagan, 2001; Wolpert et al., 1995), and action and perception are often coupled rather than separable (Friston et al., 2011). Nevertheless, such experiments could directly test whether analogous path-integration and binding operations support biological sequence modeling.

In sum, this minimal setting enables identification of candidate algorithmic components (path integration and label-position binding) that support action-conditioned sequential prediction. By retaining only the core objective of action-conditioned prediction, this minimal setting serves as a pathway for mechanistic interpretability: it narrows the space of plausible solutions while still producing nontrivial internal structure. This provides a concrete starting point for asking how such components are implemented in recurrent dynamics (or through attention in transformers), and whether analogous mechanisms arise in more complex active-vision models and in biological systems that tie together perception and action through prediction (Clark, 2013; Ha & Schmidhuber, 2018; LeCun, 2022; Nortmann et al., 2026; O'regan & Noë, 2001; Rao, 2024; Thorat et al., 2025; Wolpert et al., 1995).

Acknowledgments

The authors VB and TCK acknowledge funding by ERC StG grant 101039524 TIME. Compute resources for this project are in part funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation), project number 456666331.

We thank all members of the Kietzmann Lab at Osnabrück University, especially Philip Sulewski and Thomas Nortmann, for discussions that helped us get started on this endeavor and helped us improve the paper.

References

Banino, A., Barry, C., Uria, B., Blundell, C., Lillicrap, T., Mirowski, P., Pritzel, A., Chadwick, M. J., Degris, T., Modayil, J., et al. (2018). Vector-based navigation using grid-like representations in artificial agents. *Nature*, *557*(7705), 429–433.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, *33*, 1877–1901.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Clark, A. (2013). Whatever next? predictive brains, situated agents, and the future of cognitive science. *Behavioral and brain sciences*, *36*(3), 181–204.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273–297.

Cueva, C. J., & Wei, X.-X. (2018). Emergence of grid-like representations by training recurrent neural networks to perform spatial localization. https://arxiv.org/abs/1803.07770

Elman, J. L. (1990). Finding structure in time. *Cognitive science*, *14*(2), 179–211.

Eperon, A., Doeller, C. F., Theves, S., & Bottini, R. (2026). Action information is integrated into entorhinal representations of conceptual space and is reflected in eye movements. *Plos Biology*, *24*(4), e3003755.

Fodor, J. A., & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, *28*(1-2), 3–71.

Friston, K., Mattout, J., & Kilner, J. (2011). Action understanding and active inference. *Biological Cybernetics*, *104*, 137–160.

Greff, K., Van Steenkiste, S., & Schmidhuber, J. (2020). On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*.

Griffiths, T. L., Lake, B. M., McCoy, R. T., Pavlick, E., & Webb, T. W. (2025). Whither symbols in the era of advanced neural networks? *Trends in Cognitive Sciences*.

Ha, D., & Schmidhuber, J. (2018). World models. *arXiv preprint arXiv:1803.10122*, *2*(3).

Hummel, J. E., & Biederman, I. (1992). Dynamic binding in a neural network for shape recognition. *Psychological review*, *99*(3), 480.

Kahneman, D., Treisman, A., & Gibbs, B. J. (1992). The reviewing of object files: Object-specific integration of information. *Cognitive psychology*, *24*(2), 175–219.

Kanerva, P. (2009). Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive computation*, *1*(2), 139–159.

Kleyko, D., Rachkovskij, D. A., Osipov, E., & Rahimi, A. (2022). A survey on hyperdimensional computing aka vector symbolic architectures, part i: Models and data transformations. *ACM Computing Surveys*, *55*(6), 1–40.

LeCun, Y. (2022). A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. *Open Review*, *62*(1), 1–62.

McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, *102*(3), 419.

McCloskey, M., & Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation* (pp. 109–165, Vol. 24). Elsevier.

McCoy, R. T., Linzen, T., Dunbar, E., & Smolensky, P. (2018). Rnns implicitly implement tensor product representations. *arXiv preprint arXiv:1812.08718*.

Nortmann, T., Sulewski, P., & Kietzmann, T. C. (2026). Predictive remapping and allocentric coding as consequences of energy efficiency in recurrent neural network models of active vision. *Patterns*, *7*(1).

Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., et al. (2022). In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*.

O'regan, J. K., & Noë, A. (2001). A sensorimotor account of vision and visual consciousness. *Behavioral and brain sciences*, *24*(5), 939–973.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, *32*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, *12*, 2825–2830.

Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., & Miller, A. (2019). Language models as knowledge bases? *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th inter-*

national joint conference on natural language processing (EMNLP-IJCNLP), 2463–2473.

Plate, T. A. (1995). Holographic reduced representations. *IEEE Transactions on Neural networks*, *6*(3), 623–641.

Rao, R. P. (2024). A sensory–motor theory of the neocortex. *Nature neuroscience*, *27*(7), 1221–1235.

Scholte, H. S., & de Haan, E. H. (2025). Beyond binding: From modular to natural vision. *Trends in Cognitive Sciences*, *29*(6), 505–515.

Smolensky, P. (1990). Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial intelligence*, *46*(1-2), 159–216.

Sorscher, B., Mel, G. C., Ocko, S. A., Giocomo, L. M., & Ganguli, S. (2023). A unified theory for the computational and mechanistic origins of grid cells. *Neuron*, *111*(1), 121–137.

Thorat, S., Doerig, A., Kroner, A., Amme, C., & Kietzmann, T. C. (2025). Predicting upcoming visual features during eye movements yields scene representations aligned with human visual cortex. *arXiv preprint arXiv:2511.12715*.

Treisman, A. M., & Gelade, G. (1980). A feature-integration theory of attention. *Cognitive psychology*, *12*(1), 97–136.

Whittington, J. C., Muller, T. H., Mark, S., Chen, G., Barry, C., Burgess, N., & Behrens, T. E. (2020). The tolman-eichenbaum machine: Unifying space and relational memory through generalization in the hippocampal formation. *Cell*, *183*(5), 1249–1263.

Wolpert, D. M., & Flanagan, J. R. (2001). Motor prediction. *Current Biology*, *11*(18), R729–R732.

Wolpert, D. M., Ghahramani, Z., & Jordan, M. I. (1995). An internal model for sensorimotor integration. *Science*, *269*(5232), 1880–1882.