Hippoformer: Integrating Hippocampus-inspired Spatial Memory with Transformers

Anonymous authors
Paper under double-blind review

Abstract

Transformers form the foundation of modern generative AI, yet their key–value memory lacks inherent spatial priors, constraining their capacity for spatial reasoning. In contrast, neuroscience points to the hippocampal–entorhinal system, where the medial entorhinal cortex provides structural codes and the hippocampus binds them with sensory codes to enable flexible spatial inference. However, existing hippocampus models such as the Tolman-Eichenbaum Machine (TEM) suffer from inefficiencies due to outer-product operations or context-length bottlenecks in self-attention, limiting their scalability and integration into modern deep learning frameworks. To bridge this gap, we propose mm-TEM, an efficient and scalable structural spatial memory model that leverages meta-MLP relational memory to improve training efficiency, form grid-like representations, and reveal a novel link between prediction horizon and grid scales. Extensive evaluation shows its strong generalization on long sequences, large-scale environments, and multi-step prediction, with analyses confirming that its advantages stem from explicit understanding of spatial structures. Building on this, we introduce Hippoformer, which integrates mm-TEM with Transformer to combine structural spatial memory with precise working memory and abstraction, achieving superior generalization in both 2D and 3D prediction tasks and highlighting the potential of hippocampal-inspired architectures for complex domains. Overall, Hippoformer represents a initial step toward seamlessly embedding structured spatial memory into foundation architectures, offering a potential scalable path to endow deep learning models with spatial intelligence.

1 Introduction

The Transformer architecture has driven the recent advances in generative AI, with systems such as ChatGPT as prominent examples. This success has made the search for new architectural designs a central direction in machine learning. Transformer can be viewed as associative memories implemented through key-value caches and self-attention retrieval(Vaswani et al., 2017). However, they face inherent limitations, most notably cubic computational cost and redundant memory, which limit their scalability(Zhuang et al., 2023). To address these issues, many alternatives have been proposed by reconsidering memory design. For instance, Titans leverage fast MLP weights for large-capacity(Behrouz et al., 2024). Although these approaches improve long-sequence modeling, their memory structures remain largely flat and lack a critical element: an inherent spatial memory. Such a memory, however, is vital for organizing the "what-where" of experiences and for building internal models. Therefore, developing architectures with structured spatial memory and integrating them into modern frameworks remains an key open challenge toward efficient spatial reasoning.

Biological brains employ the hippocampal–entorhinal (HC–EC) system to construct structured spatial memory, supporting both spatial cognition and episodic memory(Buzsáki & Moser, 2013; Eichenbaum, 2017; Whittington et al., 2022). Inspired by this, many computational theories have been proposed, including CSCG(George et al., 2021; Raju et al., 2024), Tolman–Eichenbaum Machine (TEM)(Whittington et al., 2020), and Vector-HaSH(Chandra et al., 2025). As shown in Fig. 1AB, TEM provides an elegant theory in which the HC–EC

system forms a factorized memory architecture: the medial entorhinal cortex (MEC) encodes structural information through path integration, while the HC fucntions as a relational memory system to bind these structural codes with sensory codes from the lateral entorhinal cortex (LEC). This design enables generalization to novel environments and diverse task structures. Yet despite their theoretical appeal, HC–EC models have so far been validated only in simplified synthetic domains(Whittington et al., 2020; Raju et al., 2024; Zou et al., 2024; Chandra et al., 2025), and their extension to richer, real-world tasks remains an open challenge.

For example, the original TEM uses tensor-product Hebbian weights for relational memory, which is biologically plausible but computationally expensive and capacity-limited. TEM-t replaces these with key-value memory and self-attention-based retrieval, improving efficiency but still incurring high computational cost(Whittington et al., 2021). Moreover, it inherits the constraints of transformer-based architectures, such as limited context windows. Furthermore, both models demand careful memory management and parameter tuning to realize novelty-based storage. Together, these limitations hinder the practical integration of hippocampal-inspired spatial memory into modern deep learning, despite their strong conceptual motivation.

To address these challenges, we introduce mm-TEM, a scalable and efficient hippocampus-inspired structural memory, and Hippoformer, a hybrid model that integrates mm-TEM with transformers. mm-TEM introduces a novel meta-MLP memory system, meta-trained for associative binding. Building on this, Hippoformer combines this mm-TEM with transformer, yielding complementary strengths. Despite their simplicity, both models achieve strong performance on long-horizon prediction tasks in 2D and 3D environments. Our main contributions are:

1. **mm-TEM:** We propose an efficient and scalable TEM variant with a newly designed meta-MLP based relational memory. mm-TEM substantially improves training efficiency over TEM, generates grid-like patterns through self-supervised learning, and uncovers a novel link between prediction horizon and grid scales, offering insights into how different spatial grid scales are formed.
2. **Systematic evaluation:** mm-TEM is extensively tested on long sequences, large-scale environments, and multi-step prediction. It generalizes significantly better than baselines such as transformers and Titans. Ablation studies highlight the importance of the auxiliary relational loss, and further analyses show that its generalization stems from explicit understanding of spatial structures and rules, demonstrating mm-TEM as an effective structural spatial memory system.
3. **Hippoformer:** We propose Hippoformer, which integrates mm-TEM with a transformer to combine the structural spatial memory of mm-TEM with the precise working memory and abstraction capabilities of Transformer. This synergy enhances generalization in both 2D and 3D prediction tasks, demonstrating the potential of hippocampal-inspired architectures in tackling complex domains.

In summary, mm-TEM provides an efficient and scalable structural spatial memory system. And when combined with Transformer, Hippoformer has an potential to serve as a building block for enhancing spatial reasoning in deep learning.

2 Method

In this section, we present the mm-TEM and Hippoformer architecture in detail. We use the 2D grid-world prediction task as an example, where an agent moves with discrete actions $(up, down, left, right)$(Whittington et al., 2020). The input sequence is denoted as $a_0, s_0, a_1, s_1 \ldots, a_t, s_t$, where $s_t \in \mathbb{R}^d$ is the sensory observation at time t and a_t a one-hot action. The model is trained to predict the next sensory obsevation s_{t+1} given a_{t+1}, thereby mimicking hippocampal predictive coding during spatial exploration Whittington et al. (2022). The overall model structure is illustrated in Fig. 1.

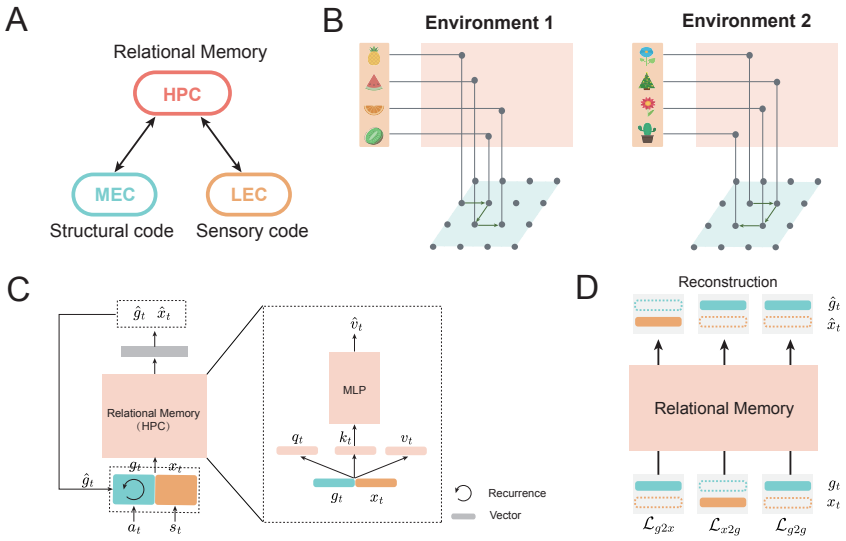

Figure 1: Factorization of structure and content in the hippocampus and model. (A) The hippocampal–entorhinal system functions as a memory system: MEC encodes structural information, LEC encodes sensory content, and HPC integrates both via conjunctive coding. (B) Structural codes in MEC can be reused across environments, enabling compositional generalization, adapted from Whittington et al. (2020). (C) The model comprises two components: a path integration network and a relational memory network, implemented as an meta-MLP memory. (D) The relational memory module is trained to reconstruct sensory codes from structural codes, structural codes from sensory codes, and both from joint inputs.

2.1 Model Architecture and Training

Following TEM theory, mm-TEM consists of two key modules: a path integration network and a relational memory network (Fig. 1C). The path integration network receives action inputs a_t and predicts the corresponding structural code g_t, while the relational memory network binds g_t with the sensory code x_t, extracted from observations s_t through a feature encoder. This design enables flexible bidirectional retrieval between structural and sensory domains.

Path Integration Network. Inspired by grid system in MEC(Moser et al., 2017), the network enforces basic spatial consistency rules (e.g., North + East + West + South = 0). Following Gao et al. (2021), we implement it as a two-layer MLP f_g with ReLU activations to map the action $a_t \in \mathbb{R}^{d_a}$ to a transformation matrix $W_t^g \in \mathbb{R}^{d_g \times d_g}$:

$$W_t^g = f_g(a_t). \tag{1}$$

The structural code is then updated as

$$\tilde{g}_t = \mathrm{ReLU}(W_t^g g_{t-1}), \quad g_t = \frac{\tilde{g}_t}{\|\tilde{g}_t\|_2}, \tag{2}$$

where ℓ_2-normalization ensures that g_t remains a unit vector.

Relational Memory Network. Mimicking hippocampal relational memory, the network binds structural and sensory codes into a joint representation $m_t = [g_t, x_t]$, enabling bidirectional retrieval between g_t and x_t. To replace the computationally expensive Hebbian weights in TEM, we introduce a meta-MLP with hierarchical fast weights Θ_t, inspired by Titans, to store relational knowledge. This design enables dynamic memorization, forgetting, and querying at test time, alleviating the complex memory management and parameter tuning required in TEM and TEM-t.

Concretely, the relational network first projects m_t into three latent vectors:

$$k_t = W_k m_t, \quad v_t = W_v m_t, \quad q_t = W_q m_t, \tag{3}$$

where k_t, v_t, q_t denote the key, value, and query representations, respectively. Rather than storing m_t directly, the meta-MLP learns to associate key k_t to value v_t online by minimizing the reconstruction loss:

$$\mathcal{L}(k_t, v_t; \Theta_t) = \left\| f_{\text{MLP}}(k_t; \Theta_t) - v_t \right\|_2^2, \tag{4}$$

where $f_{\text{MLP}}(\cdot; \Theta_t)$ denotes the meta-MLP. The fast weights Θ_t are updated by incorporating novelty-driven adaptation and forgetting:

$$\Theta_t = (1 - \alpha_t)\Theta_{t-1} + H_t, \tag{5}$$

$$H_t = \eta_t H_{t-1} - \beta_t \nabla_\Theta \mathcal{L}(k_t, v_t; H_{t-1}). \tag{6}$$

Here, $\alpha_t \in [0, 1]$ is a data-dependent gating variable that controls forgetting, paralleling hippocampal mechanisms that decay less relevant memories to preserve capacity for novel ones(Benoit & Anderson, 2012; Liu et al., 2016). The gradient $\nabla_\Theta \mathcal{L}(\cdot)$ quantifies novelty or surprisal - the mismatch between predicted and actual values - so that only unexpected inputs drive updates, akin to how hippocampus detects and prioritizes novel stimuli for long-term storage(Sinclair et al., 2021; Schomaker & Meeter, 2015). The term η_t acts as a momentum factor, averaging surprisal over a tunable timescale to stabilize learning(Bittner et al., 2017), while β_t is the learning rate. All parameters are derived from the input concatenation: $\alpha_t = \sigma(W_\alpha m_t)$, $\eta_t = \sigma(W_\eta m_t)$, $\beta_t = \sigma(W_\beta m_t)$, where σ is the sigmoid function.

The query vector q_t retrieves from memory via $f_{\text{MLP}}(q_t; \Theta_t)$, and the retrieved representation is decoded by a two-layer MLP $f_{\text{dec}}(\cdot; \Phi)$ into a joint reconstruction $\hat{m}_t = [\hat{g}_t; \hat{x}_t]$.

To explicitly enforce relational binding, we introduce three auxiliary relational losses (Fig. 1D):

(1) Structure from content: retrieve $\hat{g}_t$ given only x_t ($m_t = [\mathbf{0}, x_t]$), minimized by $\mathcal{L}_1 = \|\hat{g}_t - g_t\|_2^2$.

(2) Structure form structure: retrieve $\bar{g}_t$ given $m_t = [g_t, x_t]$, minimized by $\mathcal{L}_2 = \|\bar{g}_t - g_t\|_2^2$.

The total relational loss is

$$\mathcal{L}_{\text{rel}} = \mathcal{L}_1 + \mathcal{L}_2. \tag{7}$$

Note that other relational loss components are absorbed into the total loss, for instance, loss from structure to content.

Finally, mm-TEM incorporates a feedback loop from relational predictions bac k to the path integration network, providing error correction during navigation (Fig. 1C; see Appendix. A.2 for details).

mm-TEM Training. The objective of mm-TEM is to predict the next observation given past sensory inputs and actions. The model is trained in a self-supervised manner. During training, in the relational memory network, the projection matrices W_k, W_v, W_q and the decoder MLP weights Φ are meta-trained in the outer optimization loop, while the connection weights of the meta-MLP are optimized in the inner optimization loop. During testing, the connection weights of the meta-MLP are updated online using gradient-based update rules. We introduce a hyperparameter mb to control the memory update frequency in the relational memory network. Specifically, the connection weights in the meta-MLP are updated every mb steps. A larger mb results in sparser updates, which improves training efficiency but requires the model to rely on older information when predicting the next observation. Before training mm-TEM on downstream tasks, we perform a warm-up phase by meta-training the relational memory network with randomly generated $\{g_t, s_t\}$ sampled from a uniform distribution. This procedure, conducted for 1000 gradient steps, stabilizes training. All networks are optimized using the Adam optimizer. Additional training details and the objective loss function are provided in Appendix. A.4.

Hippformer Architecture and Its Training. Building on mm-TEM, we propose Hippoformer, which integrates mm-TEM and a one-layer Transformer in parallel to leverage the complementary strengths of both modules. In Hippoformer, the mm-TEM component is also warm-started following the same protocol described above. The model is trained using the Adam optimizer. Additional architectural and training details are provided in Appendix. A.4.

3 Results

3.1 Efficient Training and Emerge of Grid-like Representations in mm-TEM

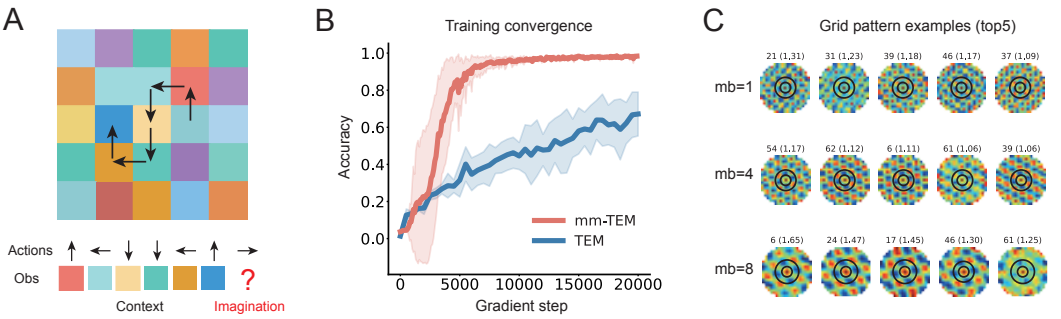

Figure 2: Task schematic and network performance. (A) A 2D Grid prediction task (example 6×6). The network predicts the next observation based on the current action at each step in sequence. (B) Multi-step test accuracy over gradient steps for mm-TEM vs TEM (same batch/sequence training length), averaged over 4 seeds. To assess generalization, the networks predicted till 256 steps from an initial 64-step context, with results averaged over four trials. More training details and parameters, see Appendix. A.1, A.2 and A.4. (C) Emerged grid scale varies with hyperparameter *mb*. Five top-gridness neurons shown per condition (more examples in Figs. A.4, A.5 and A.6).

We first ask whether mm-TEM can efficiently solve spatial reasoning task and acquire hippocampal–entorhinal-like representation. To test this, we evaluated the model on 2D grid prediction tasks (Fig. 2A)(Whittington et al., 2020), where the agent must predict the next observation based on the current action at each time step within a 256-step sequence. For each trial, the environment was sized between 9x9 and 11x11 with randomized observations, requiring the network to infer the underlying spatial structure and rules to generalize effectively. Because observations at unseen locations within this discrete environment are unpredictable, both training and evaluation are confined to positions previously encountered within each sequence.

In terms of spatial reasoning performance, mm-TEM reaches nearly 90% test accuracy within only 5,000 gradient steps, while TEM converges very slowly, achieving only about 60% accuracy even after 20,000 steps (Fig. 2B), highlighting the superior training efficiency of mm-TEM.

In terms of internal representations, analysis of the path-integration network of mm-TEM further reveals periodic grid-like representations (Fig. 2C). Notably, the grid scale is directly modulated by the update-frequency hyperparameter *mb* : larger *mb* yields coarser grids, whereas smaller *mb* produces finer scales. Since *mb* sets the effective prediction horizon, this suggests a novel mechanism for grid-scale diversity in MEC(Fyhn et al., 2004) as a naturally consequence of multi-timescale predictions in the brain.

Overall, mm-TEM not only trains efficiently but also reproduces grid-like patterns in the HC–MEC system, offering new insight into the computational basis of grid-scale diversity.

3.2 Generalization of mm-TEM in 2D grid prediction tasks

Mimicking the HC–EC system, mm-TEM acts as a structured memory that organizes knowledge for generalization. We ask: how well does such a system generalize compared to modern architectures like Transformers and Titans? To answer this, we systematically evaluate mm-TEM against these baselines in diverse settings. In the one-step imagination setting, models explore environments with varying context lengths and predict the next observation. In the multi-step imagination setting, models receive a fixed 64-step context and predict future observations conditioned on action sequences of varying lengths. These tasks probe mm-TEM's ability to generalize beyond its training horizon. Note that all models are trained and evaluated under the same task setting for fair comparison.

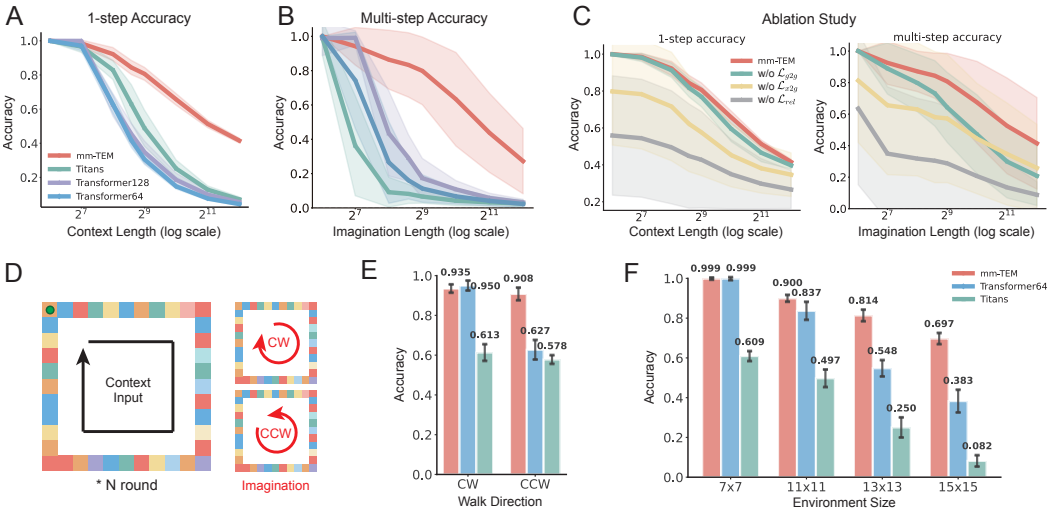

Figure 3: The Generalization and Ablation of mm-TEM. We compare mm-TEM with Titans and Transformer baselines on 2D grid prediction tasks (training length = 128). Titans uses one MAL layers. Transformer128 and Transformer64 denote 3-layer transformers with window sizes of 128 and 64 steps, respectively. (A) 1-step prediction accuracy vs context length. Networks receive an action–observation context sequence and predict the next observation. (B) Multi-step imagination accuracy vs imagination length. Networks observe a fixed 64-step context, then generate future observations conditioned on varying action lengths. (C) Ablation of auxiliary relational loss. "w/o g2g" removes $\mathcal{L}_{g2g}$, "w/o s2g" removes $\mathcal{L}_{s2g}$, "w/o rel" removes all auxiliary relational memory losses. (D) Circular-grid test setup. Networks explore an 11×11 circular environment clockwise for context, then imagine trajectories in clockwise or counterclockwise directions. (E) Clockwise vs counterclockwise performance across different architectures. (F) Effect of environment size, ranging from 7×7 to 15×15. All results are averaged over 3 seeds (see Appendix. A.3, A.4 and Fig. A.3 for more details).

In the one-step prediction task (Fig. 3A), all models perform well within the 128-step training horizon. However, Transformer and Titans rapidly degrades once the context length extends beyond this range. In contrast, mm-TEM maintains more robust performance even with sequences up to 4096 steps, retaining 40% accuracy where baselines collapse, highlighting its strong long-term generalization ability.

In the multi-step imagination task (Fig. 3B), the Transformer model with a 128-step window performs almost perfectly within its training range, but quickly drops off outside it, suggesting reliance on sequence memorization. Titans show similar behavior. In contrast, mm-TEM maintains strong long-term performance, suggesting that it has grasped the underlying spatial structure for generalization.

To determine the role of auxiliary relational loss (Fig. 3C) on this ability, we conducted ablations. Removing either $\mathcal{L}_{g2g}$ or $\mathcal{L}_{s2g}$ significantly reduces generalization ability, and eliminating all relational terms leads to severe performance degradation, confirming their importance.

Moreover, we further probe generalization under distribution shifts. In the circular-grid test (Fig. 3DE), mm-TEM achieves over 90% accuracy in the challenging counterclockwise condition, while Titans and Transformer suffer accuracy drops by up to 30%, underscoring mm-TEM's superior spatial reasoning ability. When scaling environment size from 7×7 to 15×15 (Fig. 3F) without additional training, all models decline, but mm-TEM deteriorates much more slowly and consistently outperforms the baselines. These results further show that mm-TEM generalizes beyond its training horizon, and captures spatial structure and rules more faithfully than control models, which appear to primarily rely on rote memorization.

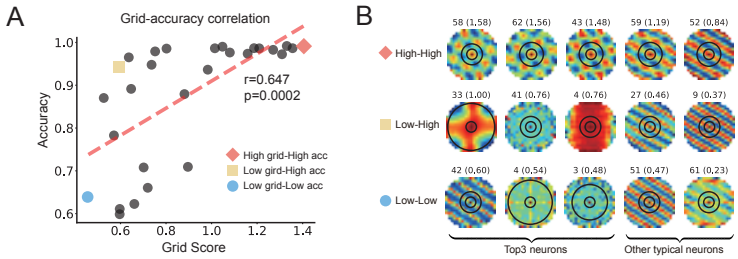

Figure 4: Long-horizon generalization and grid representations. (A) Multi-step prediction accuracy (imagination length = 512) positively correlates with grid score in mm-TEM path integration networks (r = 0.647, p = 0.0002), indicating that stronger grid-like regularity supports better generalization. (B) Representative models with high–high, low–high, and low–low grid–accuracy combinations show distinct autocorrelation patterns. For each model, the three neurons with the highest grid scores ("Top3 neurons") and other typical neurons are displayed, highlighting differences in grid-cell regularity across models. More details are provided in Appendix. A.4, Fig. A.7, A.8 and A.9.

To uncover why mm-TEM exhibits strong generalization in long-horizon inference, we examine the relationship between a model's grid score and its multi-step imagination accuracy. As shown in Fig. 4A, multi-step generalization performance in mm-TEM is closely tied to the quality of its grid-like representations. Models with higher grid scores in the path integration network consistently achieve higher prediction accuracy, suggesting that regular grid patterns facilitate long-horizon generalization.

Interestingly, we also observe cases where models with relatively low grid scores still achieve high accuracy. Visualization (Fig. 4B) reveals that these models develop alternative—but still regular—neural representations, in contrast with the unitary, unstructured patterns found in models with both low grid scores and low accuracy. Taken together, these results highlight that the presence of strongly grid-like cells is a key driver for generalization.

3.3 Hippoformer benefits from short- and long-term memory integration

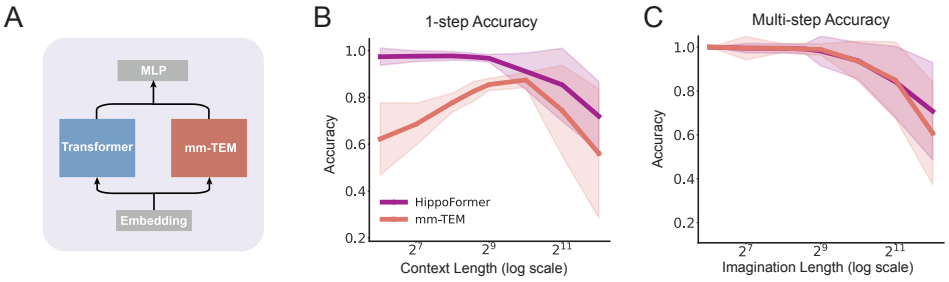

Figure 5: Hippoformer architecture and generalization in 2D grid prediction. Both Hippoformer and mm-TEM are trained using $mb = 8$ and 256-step sequences. (A) Hippoformer combines a one-layer Transformer and mm-TEM, both receiving action and sensory embeddings; their outputs are concatenated and integrated by an MLP. (B) One-step prediction accuracy of Hippoformer and mm-TEM across different context lengths. (C) Multi-step imagination accuracy across different imagination lengths, comparing Hippoformer and mm-TEM. Additional tests for effects of the memory update frequency mb are provided in Appendix. A.5 and Fig. A.2

From a memory perspective, Transformer with limited window size functions as precise short-term memory through accurate key–value caching, while mm-TEM provides a structured but less precise long-term memory. To leverage their complementary strengths, we propose Hippoformer, a unified architecture that combines a one-layer Transformer with mm-TEM. Both modules process the input embeddings independently, and their outputs are concatenated and integrated by an MLP (Fig. 5A). We evaluate all models on the 2D grid prediction task.

As shown in Fig. 5B, mm-TEM with $mb = 8$ can be trained efficiently, but its one-step prediction drops at short context lengths due to limited access to recent information, requiring longer contexts to reach strong performance. When combined with a Transformer, however, Hippoformer generalizes across both short and long context lengths. In the multi-step imagination task (Fig. 5C), where performance depends primarily on the mm-TEM component, both models achieve similar accuracy with no significant difference.

Overall, Hippoformer successfully integrates the strengths of both memory systems. The Transformer provides short-term memory for accurate short-range prediction, while mm-TEM supports structured long-horizon forecasting. This hybrid design is appealing for applications, as reducing MLP memory update frequency in mm-TEM greatly improves training efficiency and minimizes redundant memory storage, though at the cost of short-term accuracy (see Appendix. A.5 and Fig. A.2). Consequently, Hippoformer achieves both efficient training and strong generalization across diverse temporal horizons.

3.4 Hippoformer Leverages the Synergy Between Abstraction and Memorization

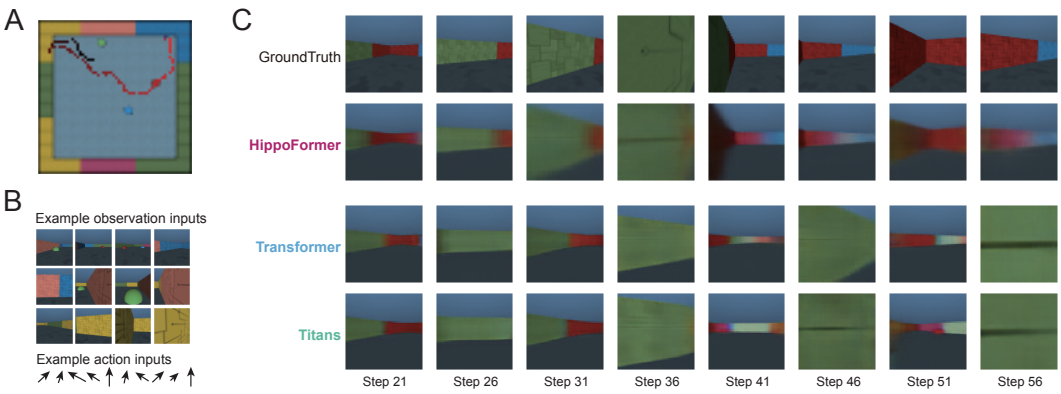

Figure 6: Hippoformer generalization in 3D environment prediction tasks. (A) Example 3D environment with randomly sampled layouts and navigation trajectories. (B) Example trajectory showing sequences of egocentric observations and actions. (C) Visualization of imagined trajectories from different models, with snapshots shown every 5 steps. More details see Appendix. A.6 and Fig. A.10 and A.11. Ablation results of the model architecture are provided in Fig. A.12 and A.13.

The hippocampus supports not only memorization but also abstraction, whereas traditional TEM and TEM-t models primarily emphasize memory storage and memory-based inference. Hippoformer bridges this gap by combining a Transformer for abstraction with mm-TEM for structured memorization, thereby integrating both capacities to enhance generalization. To evaluate this capacity, we designed a 3D empty environment task (as shown in Fig. 6AB)(Pasukonis et al., 2022). In this new setting, layout textures and egocentric trajectories are randomly sampled. Observation features are extracted through an encoder, concatenated with action inputs, and fed into the models. Each model is trained to predict the next egocentric frame over 64-step sequences. They are then evaluated on both one-step and multi-step prediction, in a manner similar to the 2D grid experiments.

Table 1: Performance comparison of different models on prediction error in 3D environments. The results are shown for both one-step and multi-step conditions, with errors reported in units of $1e-3$. The results are averaged over 3 seeds.

Models	**1-step Prediction Error (1e-3)**			**m-step Imagination Error (1e-3)**		
	Full	**Visible**	**Not Visible**	**Full**	**Visible**	**Not Visible**
Transformer	1.29±0.00	0.67±0.00	2.15±0.00	36.13±5	11.49±3.3	38.07±1.3
Titans	1.32±0.00	0.69±0.00	2.20±0.05	33.42±4.6	10.60±2.4	35.21±13
Ours	**1.27±0.00**	**0.67±0.00**	**2.09±0.05**	**9.71±0.04**	**2.72±0.01**	**10.27±3.6**

We systematically evaluate Hippoformer using three complementary metrics: accuracy across the entire sequence ("full"), accuracy on visible frames ("visible"), and accuracy on non-visible frames. Visible frames probe the model's ability to exploit structured memory, whereas non-visible frames require abstraction from historical context. As shown in Tab. 1, Hippoformer outperforms both transformer and Titans models slightly in one-step prediction but markedly in multi-step imagination. Consistent with these findings, Fig. 5C shows that Hippoformer maintains coherent predictions over long horizons in multi-step settings, whereas transformer and Titans models exhibit oscillatory errors around 36–56 steps, appearing to stack over time. Additional results are provided in Appendix A.6. Overall, these results demonstrate that Hippoformer effectively leverage abstraction and memorization, with its two modules cooperating to achieve robust long-term prediction.

4 Discussions

In this work, we introduced mm-TEM and Hippoformer, two hippocampus-inspired models for prediction and spatial reasoning. mm-TEM trains more efficiently than standard TEM and spontaneously develops grid-like codes, whose grid scale is modulated by the prediction horizon, offering a new functional perspective on grid diversity. Additionally, we propose *Hippoformer*, which integrates Transformers and mm-TEM. A natural division of labor emerges: Transformers primarily capture short-term dependencies, while mm-TEM supports long-horizon forecasting through robust grid codes in 2D environments. In 3D environments, Transformers contribute to abstraction, whereas mm-TEM focuses on memorization. Together, these complementary roles yield both improved training efficiency and stronger generalization.

Related works. Our work extends the computational theory of the HC-EC system. Existing models, such as CSCG (George et al., 2021), Vector-HaSh(Chandra et al., 2025), TEM(Whittington et al., 2021), and TEM-t(Whittington et al., 2021), are conceptually elegant but face limitations in scaling to modern deep learning architectures. For example, TEM relies on computationally expensive tensor-product Hebbian memory; TEM-t is constrained by transformer window size and requires complex memory updates; and Vector-HaSh is non-differentiable. These limitations hinder their application to complex tasks. In contrast, we propose mm-TEM, which employs a hierarchical MLP as a relational memory system. Augmented with auxiliary relational losses, mm-TEM offers a powerful, flexible memory mechanism that integrates seamlessly with modern transformers, enabling its use in more complex environments.

Long-sequence modeling is a central challenge in machine learning. Recent architectures such as Mamba(Gu & Dao, 2023), Titans(Behrouz et al., 2024), and Gated Delta Networks(Yang et al., 2024) represent important advances through structural initialization, hierarchical MLP memory, and novelty-based Hebbian rules. However, real-world information is inherently spatiotemporal, and simply enlarging memory capacity while ignoring its underlying structure is an inefficient strategy. To address this, we introduce Hippoformer, a novel hybrid memory system that combines the precise short-term memory of transformers with the structured long-term memory of mm-TEM. This design enables more efficient organization of memory sequences, making Hippoformer a promising architecture for long-sequence modeling.

Limitations and Future work. While mm-TEM provides an efficient structured memory system, our current Hippoformer design only illustrates a straightforward parallelization of transformer and mm-TEM. Moreover, the present Hippoformer is limited to a single-layer design, without leveraging the model and computation scaling that has been shown to be crucial in large language models(Kaplan et al., 2020).

Future work should investigate more efficient integration schemes and multi-layer scaling, positioning mm-TEM as a scalable fundamental building block for large systems and spatial reasoning tasks. More broadly, mm-TEM's simplicity may enable hierarchical models of the hippocampus, offering a computational handle on how biological dorsal–ventral representational gradients give rise to functional specialization(Strange et al., 2014; Maurer & Nadel, 2021).

Ethics Statement

This work does not involve sensitive datasets, human subjects, or potentially harmful applications. Therefore, we have not identified any obvious ethical concerns.

Reproducibility Statement

The paper and appendix provide detailed descriptions of model architectures, hyperparameters, training procedures, data preprocessing steps, and computational resource configurations. We also include ablation studies and additional results to ensure that the main conclusions are robust.

The Use of Large Language Models (LLMs)

We used large language models (LLMs) for minor text polishing, but not for generating scientific content, experiments, or analysis. In addition, we employed LLMs to refine visualization code for clarity and readability; these edits did not affect any experimental results.

References

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. *arXiv preprint arXiv:2501.00663*, 2024.

Roland G Benoit and Michael C Anderson. Opposing mechanisms support the voluntary forgetting of unwanted memories. *Neuron*, 76(2):450–460, 2012.

Katie C Bittner, Aaron D Milstein, Christine Grienberger, Sandro Romani, and Jeffrey C Magee. Behavioral time scale synaptic plasticity underlies ca1 place fields. *Science*, 357 (6355):1033–1036, 2017.

György Buzsáki and Edvard I Moser. Memory, navigation and theta rhythm in the hippocampal-entorhinal system. *Nature neuroscience*, 16(2):130–138, 2013.

Sarthak Chandra, Sugandha Sharma, Rishidev Chaudhuri, and Ila Fiete. Episodic and associative memory from spatial scaffolds in the hippocampus. *Nature*, 638(8051):739–751, 2025.

Howard Eichenbaum. Prefrontal–hippocampal interactions in episodic memory. *Nature Reviews Neuroscience*, 18(9):547–558, 2017.

Marianne Fyhn, Sturla Molden, Menno P Witter, Edvard I Moser, and May-Britt Moser. Spatial representation in the entorhinal cortex. *Science*, 305(5688):1258–1264, 2004.

Ruiqi Gao, Jianwen Xie, Xue-Xin Wei, Song-Chun Zhu, and Ying Nian Wu. On path integration of grid cells: group representation and isotropic scaling. *Advances in Neural Information Processing Systems*, 34:28623–28635, 2021.

Dileep George, Rajeev V Rikhye, Nishad Gothoskar, J Swaroop Guntupalli, Antoine Dedieu, and Miguel Lázaro-Gredilla. Clone-structured graph representations enable flexible learning and vicarious evaluation of cognitive maps. *Nature communications*, 12(1):2392, 2021.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Yunlong Liu, Shuwen Du, Li Lv, Bo Lei, Wei Shi, Yikai Tang, Lianzhang Wang, and Yi Zhong. Hippocampal activation of rac1 regulates the forgetting of object recognition memory. *Current Biology*, 26(17):2351–2357, 2016.

Andrew P Maurer and Lynn Nadel. The continuity of context: a role for the hippocampus. *Trends in cognitive sciences*, 25(3):187–199, 2021.

Edvard I Moser, May-Britt Moser, and Bruce L McNaughton. Spatial representation in the hippocampal formation: a history. *Nature neuroscience*, 20(11):1448–1464, 2017.

Jurgis Pasukonis, Timothy Lillicrap, and Danijar Hafner. Evaluating long-term memory in 3d mazes. *arXiv preprint arXiv:2210.13383*, 2022.

Rajkumar Vasudeva Raju, J Swaroop Guntupalli, Guangyao Zhou, Carter Wendelken, Miguel Lázaro-Gredilla, and Dileep George. Space is a latent sequence: A theory of the hippocampus. *Science Advances*, 10(31):eadm8470, 2024.

Judith Schomaker and Martijn Meeter. Short-and long-lasting consequences of novelty, deviance and surprise on brain and cognition. *Neuroscience & Biobehavioral Reviews*, 55: 268–279, 2015.

Alyssa H Sinclair, Grace M Manalili, Iva K Brunec, R Alison Adcock, and Morgan D Barense. Prediction errors disrupt hippocampal representations and update episodic memories. *Proceedings of the National Academy of Sciences*, 118(51):e2117625118, 2021.

Bryan A Strange, Menno P Witter, Ed S Lein, and Edvard I Moser. Functional organization of the hippocampal longitudinal axis. *Nature reviews neuroscience*, 15(10):655–669, 2014.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

James CR Whittington, Timothy H Muller, Shirley Mark, Guifen Chen, Caswell Barry, Neil Burgess, and Timothy EJ Behrens. The tolman-eichenbaum machine: unifying space and relational memory through generalization in the hippocampal formation. *Cell*, 183(5): 1249–1263, 2020.

James CR Whittington, Joseph Warren, and Timothy EJ Behrens. Relating transformers to models and neural representations of the hippocampal formation. *arXiv preprint arXiv:2112.04035*, 2021.

James CR Whittington, David McCaffary, Jacob JW Bakermans, and Timothy EJ Behrens. How to build a cognitive map. *Nature neuroscience*, 25(10):1257–1272, 2022.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. *arXiv preprint arXiv:2412.06464*, 2024.

Bohan Zhuang, Jing Liu, Zizheng Pan, Haoyu He, Yuetian Weng, and Chunhua Shen. A survey on efficient training of transformers. *arXiv preprint arXiv:2302.01107*, 2023.

Xiaolong Zou, Xingxing Cao, Xiaojiao Yang, and Bo Hong. Leveraging attractor dynamics in spatial navigation for better language parsing. In *Forty-first International Conference on Machine Learning*, 2024.

A Appendix

A.1 Details of 2D and 3D Prediction Tasks

2D grid prediction task. In the 2D grid environment, the agent can take four discrete actions (*up*, *left*, *right*, *down*), which are encoded as one-hot vectors. Observations are sampled from a uniform distribution over 64 sensory objects, and in each trial, the observations are drawn independently from this set. Since the environment is discrete and sensory observations are uncorrelated, generalization in this setting primarily depends on structural memory-based inference rather than feature abstraction. Consequently, predicting observations at unvisited locations is not meaningful. For all 2D grid prediction tasks, the models are trained on grid environments of sizes 8×8, 9×9, 10×10 and 11×11.

3D environment prediction task. The 3D environment is built on MemoryMaze3D (Pasukonis et al., 2022), with the layout simplified to an empty 2D plane. Environment textures are randomly sampled in each trial. Unlike the allocentric observations in the 2D grid environment, the observations here are egocentric view images, which are inherently more complex. The action space consists of a discrete set of actions, including moving forward, turning left, turning right, staying still, and combinations of moving forward with turning. However, the resulting movements are continuous, subject to noise and acceleration. In this environment, unvisited observations can be inferred from nearby spatial information, making feature abstraction a critical factor in the 3D environment prediction tasks.

A.2 Details of mm-TEM Architecture

The mm-TEM architecture primarily consists of a path integration network and a relational memory network. Unlike the relational memory in TEM and TEM-t, which incorporate hand-designed relational priors - TEM uses a tensor-product-based Hebbian mechanism, and TEM-t employs key-value pairs - our relational memory does not rely on any manually injected priors. Instead, we self-supervise the network with a simple auxiliary relational loss, allowing the memory system to learn relational structure autonomously. This approach makes the memory system more flexible and scalable compared to those in TEM and TEM-t.

mm-TEM receives both action vectors and sensory feature vectors as inputs. In the 2D grid environment, sensory features are extracted using a 2-layer MLP encoder, whereas in the 3D environment they are extracted using a 4-layer convolutional neural network and 1-layer MLP. The path integration network processes the action sequences and generates structural codes. Due to noise-driven error accumulation, the path integration network requires correction signals from sensory observations to recalibrate the structural codes, analogous to the HC–EC system in biological brain. Further error correction details are provided below.

Error correction in the path integration system. In the HC–EC system, visual sensory cues provide feedback from the HC to correct path integration errors in the MEC. To emulate this mechanism, we introduce a feedback loop from the relational memory network. Specifically, given an action a_t, the path integration network generates a structural code $g_{gen,t}$. Together with the sensory code x_t, a memory query $m_t = [g_{gen,t}; x_t]$ is formed and sent to the relational memory network, which retrieves a structural code $\hat{g}_{gen,t}$ and a content code $\hat{x}_t$. The retrieved code is then fed back into the path integration network and combined with $g_{gen,t}$ as follows:

$$g_{inf,t} = g_{gen,t} + \alpha(\hat{g}_{gen,t} - g_{gen,t}) \cdot f_{delta}(g_{gen,t}, \hat{g}_{gen,t}, ||x_t - \hat{x}_t||^2)$$

where $f_{delta}(\cdot)$ is two-layer MLP that predict the variance of the integrated structural code, respectively. The scalar α controls the integration ratio. The updated structural code, $g_{inf,t}$, then becomes the new state of the path integration network and is stored in the relational memory alongside the sensory code, x_t.

Objective Functions and Training. The overall prediction and update process in mm-TEM is summarized as follows:

1. Given the action a_t, the path integration network computes the structural code $g_{gen,t}$.
2. With the input $m_t = [g_{gen,t}; \mathbf{0}]$, the relational memory network retrieves the sensory code, which is further decoded into the observation $\hat{s}_{gen,t}$. This yields the *generative prediction loss*:
$$\mathcal{L}_{\text{gen}} = \|\hat{s}_{gen,t} - s_t\|_2^2.$$
3. Using the joint structural–sensory code $m_t = [g_{gen,t}; x_t]$, the relational memory network predicts the corrected structural code $\hat{g}_{gen,t}$. Combining the generative and feedback-retrieved structural codes $(g_{gen,t}, \hat{g}_{gen,t})$, mm-TEM produces the corrected structural code $g_{inf,t}$ using the above error correction process. This gives the *consistency loss*:
$$\mathcal{L}_{\text{con}} = \|g_{gen,t} - g_{inf,t}\|_2^2.$$

4. The corrected structural code $g_{inf,t}$ is then passed again to the relational memory network with $m_t = [g_{inf,t}; \mathbf{0}]$. The network predicts the sensory code $\hat{x}_{inf,t}$, which is decoded into the observation $\hat{s}_{inf,t}$. This defines the *inference prediction loss*:

$$\mathcal{L}_{\text{inf}} = \|\hat{s}_{inf,t} - s_t\|_2^2.$$

5. Finally, the integrated structural code $\hat{g}_{inf,t}$ and the sensory code x_t are stored in the relational memory network using the online gradient-based update rule.

The main objective of mm-TEM is to predict the next observation given past sensory inputs and actions. It is trained via self-supervised manner. The total loss combines the predictive loss, consistency loss and auxiliary loss:

$$\mathcal{L} = \gamma_{rel}\mathcal{L}_{\text{rel}} + \gamma_{gen}\mathcal{L}_{\text{gen}} + \gamma_{con}\mathcal{L}_{\text{con}} + \gamma_{inf}\mathcal{L}_{\text{inf}}. \tag{8}$$

where $\gamma_{rel}, \gamma_{gen}, \gamma_{con}$ and γ_{inf} are scale factor of different losses. We train the entire mm-TEM end-to-end on the prediction task by minimizing $\mathcal{L}$. Optimization uses the Adam optimizer with a learning rate of 0.001, training is run for up to 20,000 steps.

When combined with Transformer, the model architecture of Hippoformer is shown in a more detaied version of Fig. 2A in Fig. A.1.

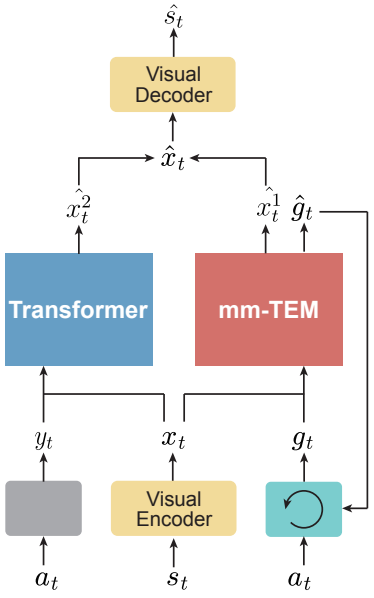

Figure A.1: Model architecture of Hippoformer, which integrates mm-TEM with a Transformer in parallel. Related to Fig. 2A

A.3 Control model architecture

We compare mm-TEM and Hippoformer with two control architectures: Transformers and Titans. The model configurations are summarized as follows.

For the Transformer baseline, we employ a three-layer Transformer with two different temporal windows: 64 steps and 128 steps. The number of attention heads is set to $N_{\text{head}}^T = 8$ and the hidden dimension to $N_{\text{dim}}^T = 64$. Although the Transformer with a 64-step window has limited temporal context, its hierarchical three-layer structure still allows it to capture dependencies across the entire sequence.

For Titans, we adopt the MAC architecture, which incorporates an external memory component as contextual information. The meta-MLP is parameterized by the number of layers ($D_{mem} = 2$), the number of attention heads ($N_{head}^m = 2$), and the hidden dimension

($N^m_{dim} = 64$). Its Transformer component uses the same parameterization as the standard 64 windows Transformer. Both the Transformer and Titans share the same input layer, which embeds actions and sensory inputs into $N_{act} = 192$, $N_{sense} = 192$ in 2D environments, and $N_{act} = 128$, $N_{sense} = 4096$ in 3D environments.

For our models (mm-TEM and Hippoformer), we use 64 grid cells for 2D environments and 128 grid cells for 3D environments. The meta-MLP parameters are identical to those used in Titans. The short-term memory module is implemented as a one-layer Transformer with a window size of 32. The input layer embeds grid representations into $N_{grid} = 64$, $N_{sense} = 64$ in 2D environments, and $N_{grid} = 128$, $N_{sense} = 4096$ in 3D environments.

A.4 Training and Parameter Details

All models are implemented in `PyTorch` and trained on NVIDIA A100 GPUs. The training and parameter configurations corresponding to each figure are summarized below.

For the results in Fig. 2, we report the sequence length, learning rate, batch size, memory update frequency (mb), and testing environment size. Detailed settings are provided in Table A.1.

Table A.1: Training and parameter details for different experimental results.

Figure	Sequence Length	Learning Rate	Batch Size	mb	Env. Size
Fig. 2	256	1e-3	16	8	8-11
Fig. 3	128	1e-3	16	1	8-11
Fig. 4	256	1e-3	16	1/4/8	8-11
Fig. 5	256	1e-3	16	8	8-11
Fig. 6	64	5e-4	16	8	9
Fig. A.3	128	1e-3	16	1	8-11
Fig. A.2	128	1e-3	16	1/4/8	8-11

A.5 Effects of hyperparameter mb on mm-TEM

In mm-TEM, the meta-MLP memory is updated every mb steps; thus, the hyperparameter mb controls the memory update frequency. With a larger mb, the network must rely on information from mb steps ago to predict the next observation, effectively increasing the prediction horizon. Therefore, mb also balance short- and long-range predcitions. As shown in Fig. **??**AB, under one-step prediction, smaller mb values emphasize short-range predictions but perform worse in multi-step prediction.

Ideally, models should perform well in both short- and long-range prediction and generalization. This is where Hippoformer demonstrates its advantage: by combining short-term memory from the Transformer with a limited window size and structured long-term memory from mm-TEM, Hippoformer achieves strong performance across both short- and long-range predictions, as shown in Fig. 5 in the main text. Furthermore, larger mb also improves training efficiency for mm-TEM. Taken together, this simple combination leverages the complementary strengths of both modules while also achieving high efficiency.

A.6 Other supplementary results

Other supplementary results corresponding to the Results in the main text are shown as follows.

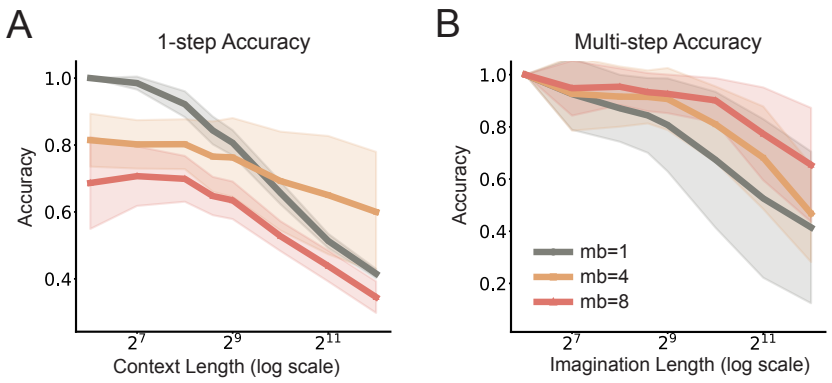

Figure A.2: Effects of Memory Update Frequency (mb) on Generalization Capacity of mm-TEM. We investigate how the memory update frequency parameter (mb) affects the generalization capacity of mm-TEM. All models are trained with a sequence length of 128. The generalization performance is evaluated under both one-step prediction (A) and multi-step prediction settings (B).

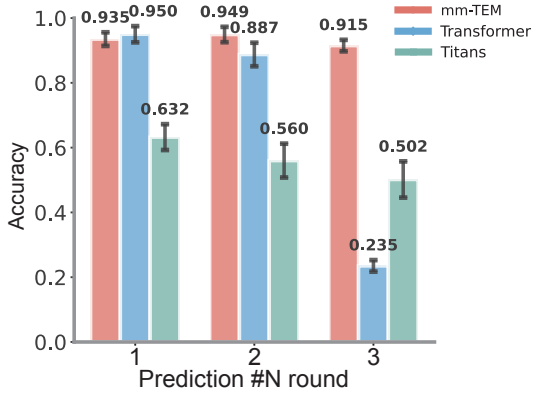

Figure A.3: Generalization Capacity of mm-TEM in Long-Context Conditions. we evaluates the generalization capacity of mm-TEM on the n-round imagination task in a circular grid environment, the same as that in Fig. 3D. As the agent explores the environment in a clockwise manner and accumulates more rounds of sensory experience, mm-TEM achieves robust performance, maintaining around 90% accuracy in multi-step imagination. In contrast, both Transformers and Titans exhibit a sharp performance drop as the context input length increases, highlighting the superior generalization ability of mm-TEM under long-context conditions.

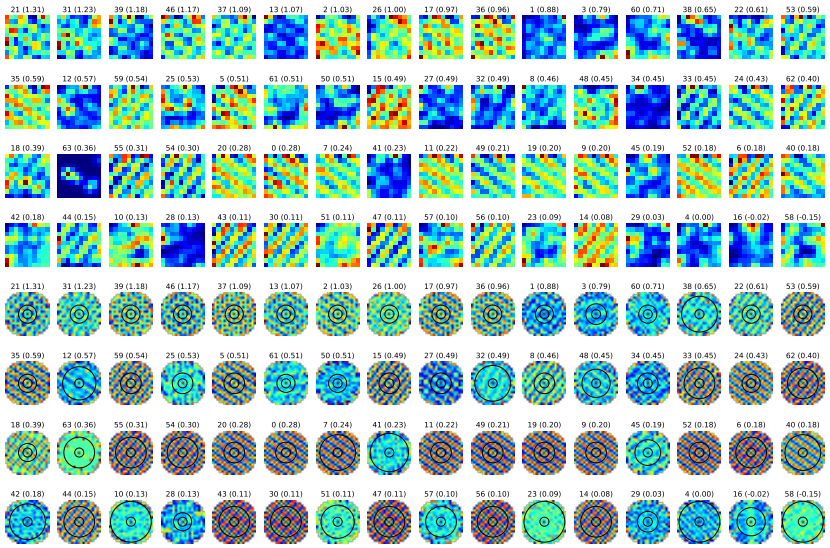

Figure A.4: Grid patterns from mm-TEM with parameter mb=1. Related to Fig. 2C.

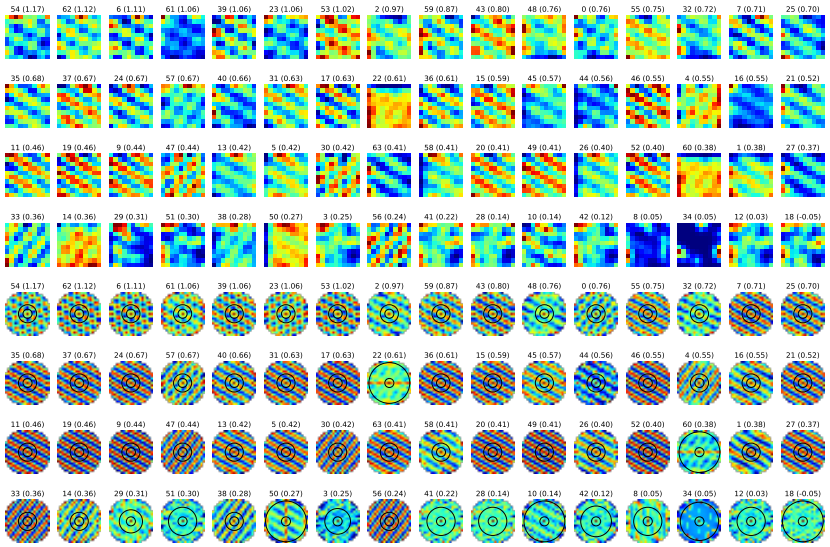

Figure A.5: Grid patterns from mm-TEM with parameter mb=4. Related to Fig. 2C.

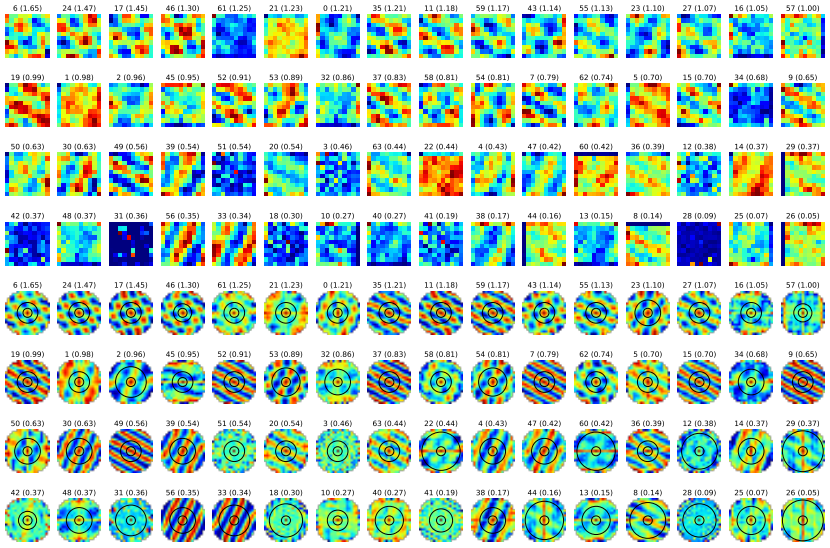

Figure A.6: Grid patterns from mm-TEM with parameter mb=8. Related to Fig. 2C.

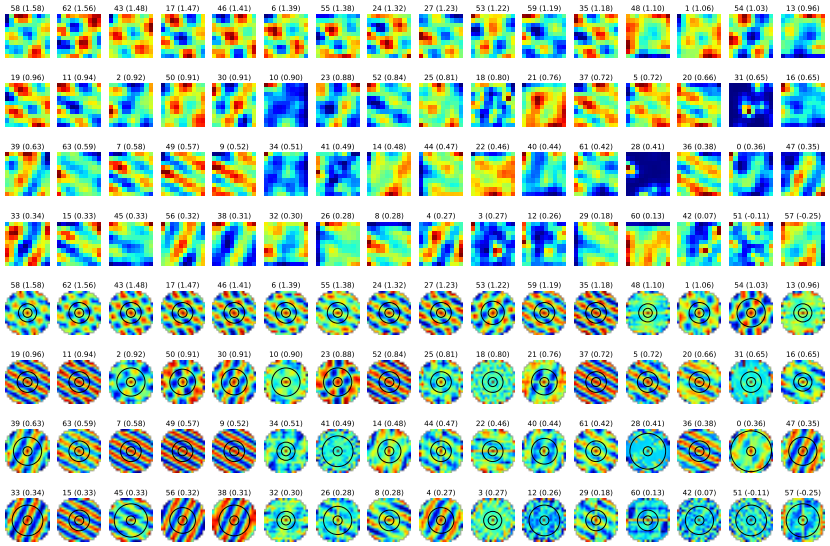

Figure A.7: Grid patterns in representative mm-TEM model whose gridness and accuracy are high. Related to Fig. 4B.

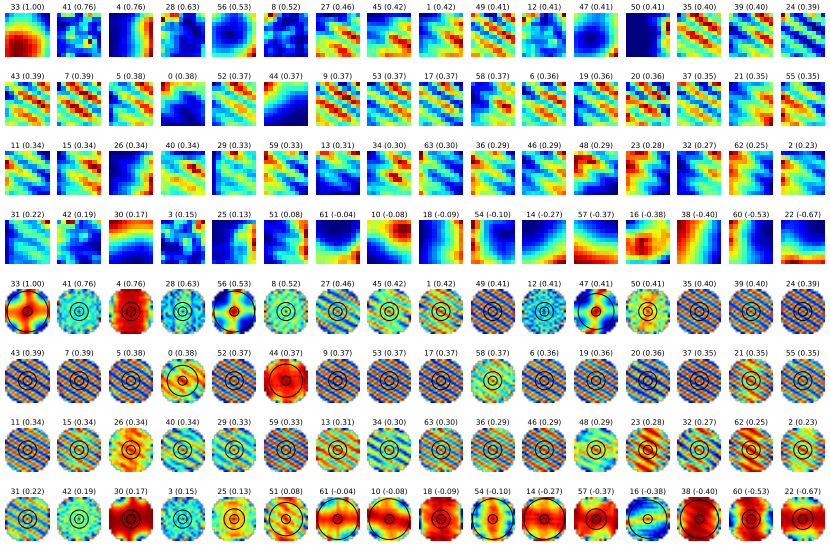

Figure A.8: Grid patterns in representative mm-TEM model whose gridness is low and accuracy is high. Related to Fig. 4B.

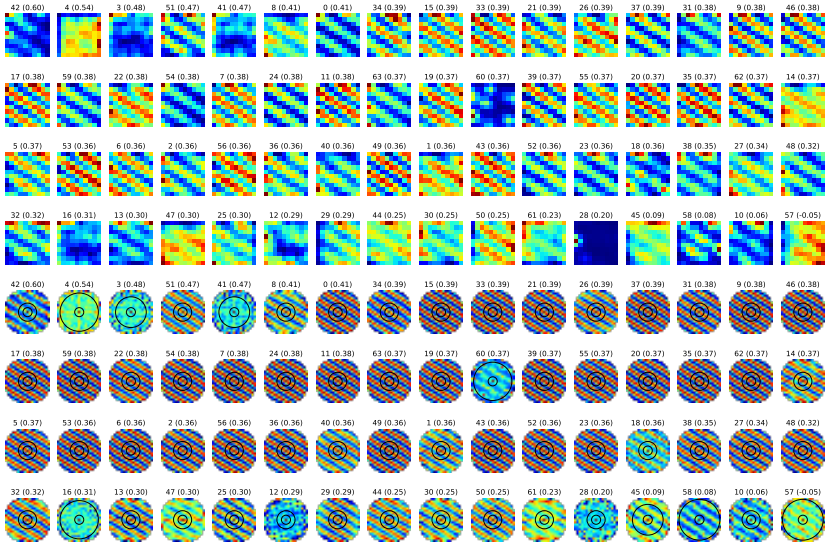

Figure A.9: Grid patterns in representative mm-TEM model whose gridness and accuracy are low. Related to Fig. 4B.

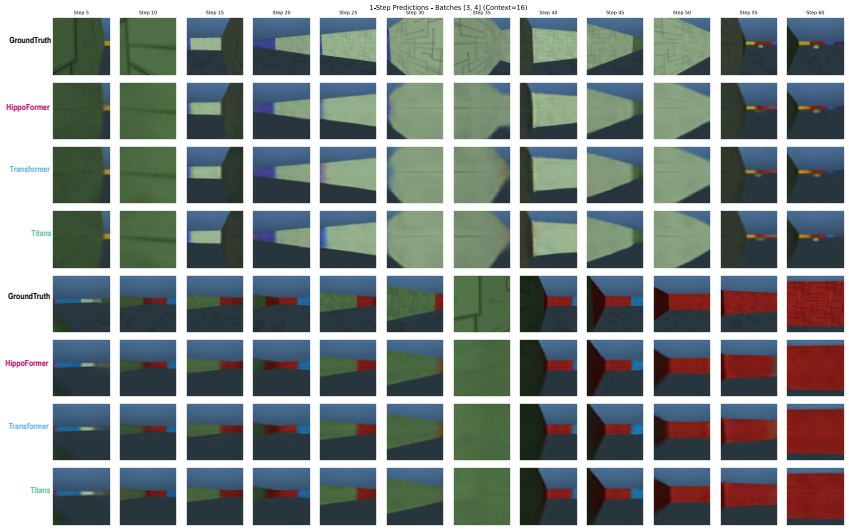

Figure A.10: Visualization of example trajectories of one-step prediction in two 3D environments from different models, with snapshots shown every 5 steps. Related to Fig. 6C.

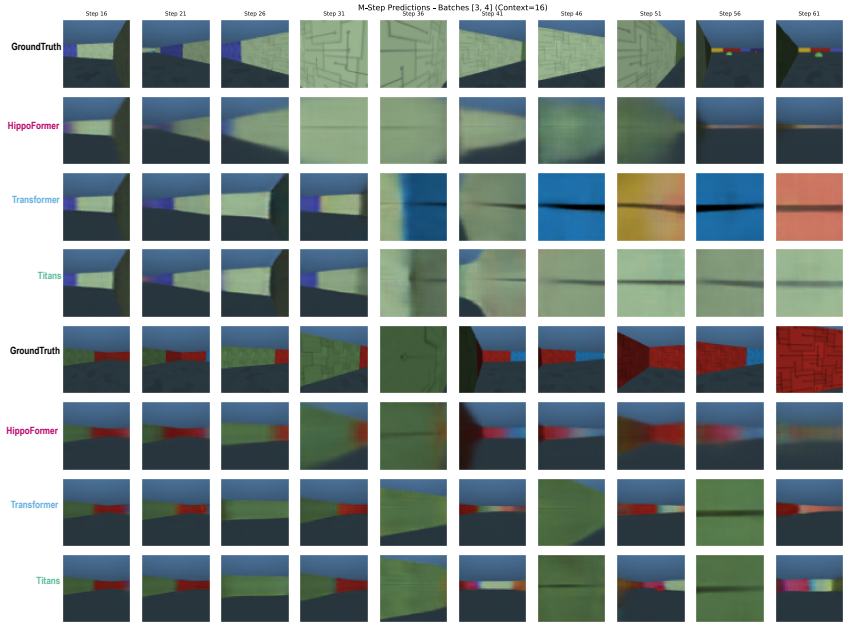

Figure A.11: Visualization of example trajectories of multi-step prediction in two 3D environments from different models, with snapshots shown every 5 steps. Related to Fig. 6C.

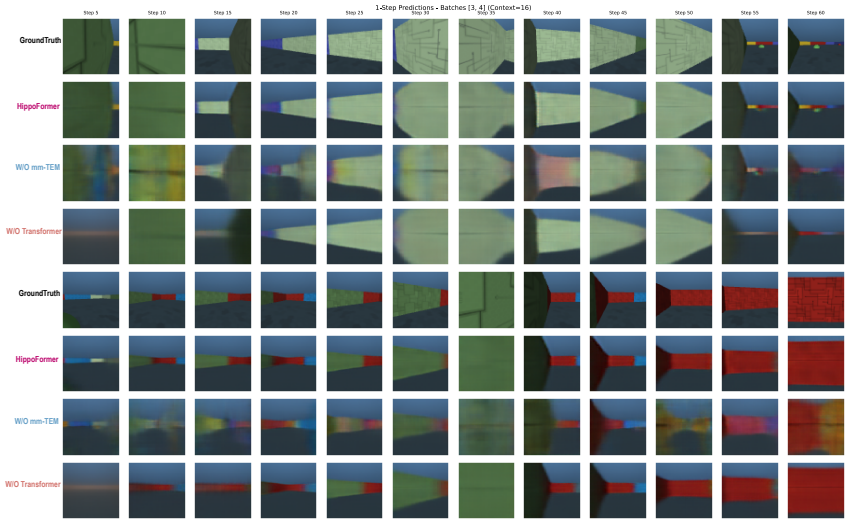

Figure A.12: Visualization of example trajectories of one-step prediction in two 3D environments from Hippoformer with ablation, with snapshots shown every 5 steps. Related to Fig. 6.

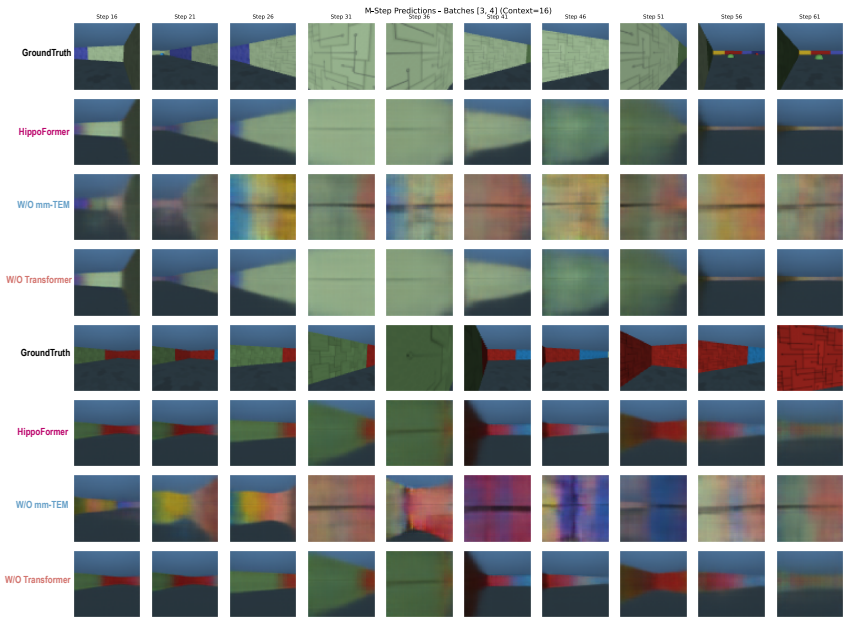

Figure A.13: Visualization of example trajectories of one-step prediction in two 3D environments from Hippoformer with ablation, with snapshots shown every 5 steps. Related to Fig. 6.