SPATIALLY-AWARE TRANSFORMER MEMORY FOR EMBODIED AGENTS

Junmo Cho[1*], **Jaesik Yoon**[1,2*], **Sungjin Ahn**[1]
[1]KAIST & [2]SAP

ABSTRACT

Episodic memory plays a crucial role in various cognitive processes, such as the ability to mentally recall past events. While cognitive science emphasizes the significance of spatial context in the formation and retrieval of episodic memory, the current primary approach to implementing episodic memory in AI systems is through transformers that store temporally ordered experiences, which overlooks the spatial dimension. As a result, it is unclear how the underlying structure could be extended to incorporate the spatial axis beyond temporal order alone and thereby what benefits can be obtained. To address this, this paper explores the use of Spatially-Aware Transformer models that incorporate spatial information. These models enable the creation of place-centric episodic memory that considers both temporal and spatial dimensions. Adopting this approach, we demonstrate that memory utilization efficiency can be improved, leading to enhanced accuracy in various place-centric downstream tasks. Additionally, we propose the Adaptive Memory Allocator, a memory management method based on reinforcement learning that aims to optimize efficiency of memory utilization. Our experiments demonstrate the advantages of our proposed model in various environments and across multiple downstream tasks, including prediction, generation, reasoning, and reinforcement learning. The source code for our models and experiments will be available at https://github.com/spatially_aware_transformer.

1 INTRODUCTION

Episodic memory is the type of memory that involves storing and retrieving events or episodes from an individual's life. This memory plays a crucial role in various human cognitive processes, such as enabling people to mentally travel back in time and remember past experiences like "what you did yesterday" (Perrin & Michaelian, 2017; Suddendorf et al., 2009). It is also foundational in imagining the future and making effective decisions (Schacter et al., 2017). Therefore, AI systems aiming to mimic human-like cognitive abilities would benefit from incorporating a form of episodic memory.

Currently, the primary approach for modeling episodic memory in AI is through transformers (Parisotto et al., 2020; Lampinen et al., 2021). This approach represents an episodic memory as a sequence of experience frames. During the storing process, frames are stored in First-In-First-Out (FIFO) order with temporal-order embedding, called positional embsedding. While this FIFO inductive bias seems reasonable in modeling episodic memory due to the uni-directional flow of time in the physical world, various studies in cognitive science suggest that the formation and retrieval of episodic memories rely not only on the temporal aspects but also on another foundational axis of the physical world, the spatial axis, such as where we parked our car this morning (Buzsáki & Tingley, 2018; Ekstrom & Ranganath, 2018; Pathman et al., 2018; van Asselen et al., 2006).

Despite this significance, the spatial dimension has largely been overlooked in the modeling of transformer-based episodic memory. This can be attributed to two main reasons. First, since their invention, transformer models have mainly advanced in the language domain where spatial dimensions are not as prominent or meaningful. Second, in some domains, spatial annotations, such as the position of a robot, may not be as easily accessible as the temporal order indices. As a result, previous research have concentrated mainly on improving the accuracy of inferring such spatial

*Equal contribution. Correspondence to junmokane@kaist.ac.kr and sjn.ahn@gmail.com.

information from experience, e.g., sequences of observations and actions (Fraccaro et al., 2018; Uria et al., 2020; Gregor et al., 2019; Whittington et al., 2022).

However, beyond language, there are a broad range of domains, such as robotics and virtual agents in games, where the spatial dimension is just as natural and crucial as the temporal dimension. These agents need to navigate around rooms and various areas, and having spatially structured knowledge such as "*what happened where*" is essential for achieving the goal. Moreover, in many cases, the spatial information is as easily accessible as the time indices. In virtual worlds such as games, for instance, the ground-truth position information is easily provided from the game engine (Guss et al., 2019) and for robots, it is already popular to provide such spatial information by a separate hardware system such as BLE beacons (if it is in-door) (Faragher & Harle, 2015), GPSs (El-Rabbany, 2002), or algorithmically, e.g., SLAM (Taketomi et al., 2017; Zhang et al., 2017). Despite these circumstances, transformers currently only consider time axis.

In this work, we pose the question: *what if spatial information is available as time indices to transformer-based models?* Our goal is to investigate this overlooked aspect of transformers and to examine whether and how transformers can utilize the spatial axis. We approach this by deviating from the traditional focus on acquiring spatial information. Considering the widespread use of transformer architectures, as well as the growing accessibility of spatial information for embodied agents, we believe it is appropriate and timely to address this question.

To achieve this goal, we propose a series of transformer architectures called Spatially-Aware Transformers (SAT or SA-Transformer), which incorporate spatial information into transformers. This allows us to explore various feasible design choices and assess the pros and cons of these architectures. Specifically, we demonstrate that Spatially-Aware Transformers offer three main benefits. Firstly, they **enhance spatial reasoning** tasks. We show that without explicit spatial information (only with temporal information), it is challenging to solve various general spatial reasoning tasks, such as "What has happened in the next room of the agent?" (as shown in Figure 1 (a)). However, providing spatial information makes these tasks much easier. Secondly, they enable **efficient memory management**. By incorporating spatial information, we can overcome the limitations of the FIFO memory, such as deleting important memory solely because it is old (as shown in Figure 1 (b)). We can also implement structured approaches like place-centric hierarchical memory. We demonstrate that this structured memory improves effectiveness and efficiency in various place-centric tasks. To enhance memory utility further, we introduce the Adaptive Memory Allocator (AMA). Unlike traditional transformers that follows the FIFO policy for memory writing, AMA allows for adaptive selection of various writing strategies based on the goal. Finally, we demonstrate that these **benefits extend** across a wide range of machine learning problems, including supervised prediction, image generation, and reinforcement learning.

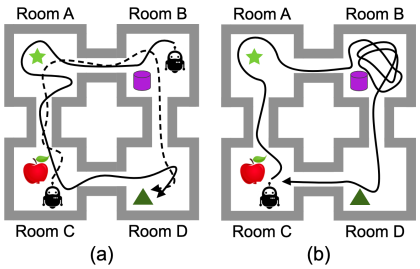
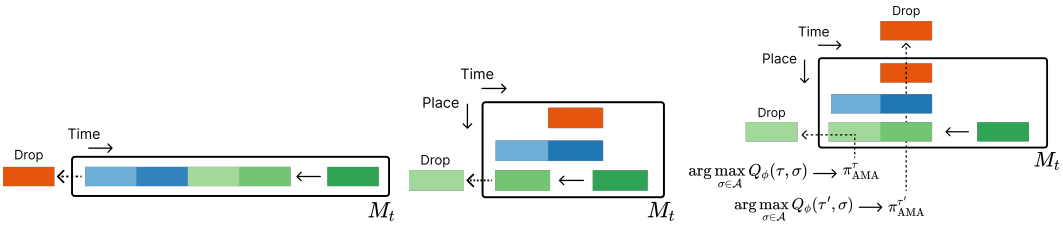


Figure 1: Home robot thought experiments. In scenario (a), a robot visits the environment in a different temporal order. When it arrives at Room D, answering what happened to the room on its left would not be easy using only the temporal axis. In scenario (b), the robot stays in Room B for a long time. With a FIFO memory, the agent would delete the memory of Room C. When it returns, it will perceive Room C as new.

The main contributions of the paper are as follows: First, to our knowledge, we are the first to motivate, conceptualize, and introduce the notion of transformers capable of utilizing explicit spatial information. Second, we propose a series of SAT models designed to leverage spatial annotation. Third, we introduce the AMA method, which enables adaptable selection of various memory management strategies beyond the traditional FIFO-only strategy. Finally, we demonstrate that the advantages of SAT are not limited to a specific problem domain but extend across an array of machine learning applications, including supervised prediction, image generation, and reinforcement learning. We also release the source code for reproducibility.

2 Proposed Models

In this section, we describe the proposed Spatially-Aware Transformer (SAT) models. We begin with the simplest model, SAT-FIFO, and then discuss incorporating Place-Memory into SAT. Lastly, we



(a) FIFO Memory **(b)** Place Memory **(c)** Adaptive Memory Allocator (AMA)

Figure 2: Illustrations of FIFO, Place Memory, and the Adaptive Memory Allocator (AMA). The orange memory is the oldest and originates from a different place compared to the green and blue ones. In FIFO, the orange memory is dropped due to its age. However, the Place Memory system retains it through place-wise memory allocation. AMA chooses the appropriate allocation strategy, π^{τ}_{AMA}, based on the given downstream task knowledge, denoted as τ.

introduce Adaptive Memory Allocation as a method for selecting a strategy other than FIFO based on the purpose of the downstream tasks.

2.1 Design 1: SAT with FIFO Memory

When location annotation ℓ_t is available at each time step t, the most direct approach to make a spatially-aware episodic memory is simply to introduce the spatial embedding $e_t^{loc} = \texttt{embed}(\ell_t)$ and combine it with the time embedding[1] e_t^{time} and the observation embedding e_t^{obs}. For the time and spatial embedding, the learnable embedding (Devlin et al., 2018) or sinusoidal positional embedding (Vaswani et al., 2017) could be applicable, but in this literature, for simplicity, the sinusoidal positional embedding is used. This results in the experience frame $x_t = \texttt{sum_embed}_x(e_t^{loc}, e_t^{time}, e_t^{obs})$. Then, we add the experience frame to the episodic memory M_t in the conventional FIFO order. We call this simplest model SAT-FIFO and its memory structure is illustrated in Figure 2a.

The main advantage of this approach is that it requires minimal alteration from the standard FIFO transformer in implementing the three basic operations of an episodic memory: `WRITE` ($M_{t+1} = \texttt{Write}(M_t, x_t)$), `READ` ($y_t = \texttt{Read}(M_t, q_t)$), and `REMOVE` ($M_{t+1} = \texttt{Remove}(M_t, \sigma)$). Here, q_t represents the query used to retrieve information from the episodic memory, and y_t is the output of the retrieval operation that will be used to solve downstream tasks. When it comes to removal, we generalize the operation by introducing a strategy σ that determines how to remove experience frames. In the case of SAT-FIFO, the strategy σ is fixed to FIFO. Although this approach endows the standard transformer with spatial awareness with minimal effort, it has limitations. For example, when the memory is full, the SAT-FIFO model has to remove the oldest experience first even if it is important to keep it for achieving the goal of the downstream task, e.g., as illustrated in Figure 1 (b).

2.2 Design 2: SAT with Place Memory

To resolve the limitations of SAT-FIFO, we introduce the Place Memory (PM) into the Spatially-Aware Transformers. In this model, called SAT-PM, the whole space is represented by a number of places and we maintain a place-wise episodic memory M_t^k one per place $k = 1, ..., K$, resulting in a place-centric hierarchical episodic memory $M_t = \{M_t^k\}_{k=1}^K$, as shown in Figure 2b.

A place can be viewed as a cluster of locations and we observe that this place structure is naturally provided in many applications, e.g., via BLE beacons for indoor robots, via GPS and maps for autonomous driving cars, and via clustering algorithms in general. Even if the place structure must be learned through a clustering algorithm, we note that it is a relatively simple learning task due to the extremely low dimensionality (only 2 or 3) of the spatial coordinates. Moreover, as we shall show in Exp-5 in Sec. 3.2, it is not always necessary to have the ground truth place structure but an approximate one can be adequate in practice. Thus, in our experiments, we assume that the location is already clustered and thus replace the location embedding with place embedding e_t^{place}.

Place-Centric Store: To store the experience frame x_t^k obtained at place k, it is added to M_t^k in a FIFO order. The length of each place's episodic memory, L_p, can be set to L/K by default or

[1]Technically, it is not time embedding, but *position* (or temporal order) embedding. However, for simplicity, we refer to it as time embedding throughout the paper because "position" can also refer to spatial position.

manually adjusted with the prior's guidance. This allows the Place Memory to retain the memory of a room, regardless of how long ago it was visited, as long as the place's memory is not full. Unlike the flat FIFO transformer, this approach provides greater flexibility, balancing total memory capacity between time and space and help mitigate the issue shown in scenario (b) in Figure 1. However, if a place's memory is full, experience frames will still be removed in a FIFO order. Although such deletions are inevitable in any episodic memory with finite capacity, in the next subsection, we will discuss different memory management strategies that can handle this situation better than FIFO.

Place-Centric Hierarchical Read: A critical issue with transformers is their computation cost increasing with the number of memory items (Vaswani et al., 2017). One approach to mitigate this problem in an episodic transformer is to group a set of experience frames (Lampinen et al., 2021) into *chunks* and perform hierarchical read operations. Specifically, we first select the top-k chunks by applying attention only across the chunk representations instead of individual experience frames. Then, we search for experience frames within those chosen chunks, as shown in Figure 9.

Previously, only time-based chunking was available. However, we observe that this approach can lead to problems in spatial environments. This is because experience frames of a place can be scattered across chunks located at various positions in the flat memory (depending on the visit order of a trajectory), and similarly, a chunk can contain mixed experiences from multiple places. This makes it more difficult to learn to search for these frames and develop a representation of what happened in a place. Utilizing spatial awareness, our Spatially-Aware Transformer provides a simple solution to address this issue. We first ensure that a chunk is filled only with experiences of the same place, and these chunks are managed within a place memory. In experiments, we show that this simple place-centric hierarchical reading is both fast and effective for spatial reasoning. For more details about the hierarchical operation, please refer to the Appendix B.2.

2.3 Design 3: Adaptive Memory Allocator

So far, we have assumed that experience frames are added to each place memory in the FIFO order. While using the inductive bias, "*recent knowledge is more important*", is reasonable in modeling languages, it may not hold for various spatial problems. For example, a task may simply require retaining memory of a place that occurred at the beginning of an episode. Hence, to have a generic episodic memory, a more desired approach is to decide what to remember or forget *adaptively* according to the goal of the task.

However, a dilemma exists in practice. If we prioritize flexibility, the best approach would likely be to learn the optimal decision of which knowledge to be updated *at each time step* (Graves et al., 2014; 2016; Hung et al., 2019). However, it is known that this approach is notoriously difficult to train in practice, and we show it empirically in Appendix D.6. In fact, transformers completely remove the problem of learning to write into the memory by adopting the FIFO policy. This is one of the most crucial advantages of the transformer model, which allows it to be scalable in the language domain but at the cost of losing adaptability to find a better writing strategy.

To address this issue, we propose a simple and practical method called Adaptive Memory Allocator (AMA) that balances flexibility of the writing policy and ease of training. AMA extends the space of memory management strategies to multiple strategies by allowing the developer to design the space of various memory management strategies, denoted as $\mathcal{A} = \{\sigma_1, \ldots, \sigma_M\}$, based on their prior knowledge. For instance, σ_1 can be FIFO, σ_2 can be Last-In-First-Out (LIFO), σ_3 can be Least-Visited-First-Out (LVFO), and so on. This way, the traditional transformer can be viewed as a special case where there is only one strategy, i.e., FIFO. Also, these strategies are similar to CPU-cache management algorithms (Denning, 2005), but our goal is to learn how to choose between them. It is important to note that the primary value of this method lies not in what the strategies themselves are, as they can be created in various ways based on prior knowledge. Rather, the value lies in the fact that Spatially-Aware transformers enable the selection of these strategies (other than FIFO) in a way that optimizes downstream tasks.

The problem is then formulated as choosing the best strategy based on the agent's current intention or task description τ. For example, the agent can be given $\tau =$ "*What object did you see in your left room?*" This problem can be addressed by learning a policy $\pi(\sigma|\tau)$. We implement this as a one-step Q-learning policy $\pi_{\text{AMA}} = \arg\max_{\sigma \in \mathcal{A}} Q_\phi(\tau, \sigma)$, where ϕ is the parameter of the value function. One might think that, if the number of strategies is small enough, we can simply try all strategies and

pick the best one. Then, we do not need to learn how to choose one. However, as there are almost infinitely many ways of describing a task, e.g., via language, it should be able to generalize to the variations of task descriptions. By learning the context-aware policy in AMA, we can learn to choose a suitable strategy even if an unseen (as a variation of) task description is given at test time. AMA is illustrated in Figure 2c, and the pseudo-code for learning algorithm with SAT and AMA is presented in the Appendix B.3.

3 EXPERIMENTS

This section presents an empirical evaluation of the Spatially-Aware Transformer (SAT) and Adaptive Memory Allocator (AMA) across various environments and downstream tasks. We begin by evaluating the models on prediction tasks in the Room Ballet environment. Then, we demonstrate its capability to build an action-conditioned world model and spatially-aware image generation model. Lastly, we showcase the use of SAT-AMA for training a downstream reinforcement learning policy. Each baselines and tasks are explained in each of the experiment section. In the experiments, observation, the time-step, and space are sequentially presented to the agent. Afterward, a query is provided. For some experiments, the task type is introduced at the episode's commencement to select the proper memory allocation strategy. The agent is expected to integrate these inputs to solve the tasks. The more details are in Appendix C.1.

3.1 SUPERVISED PREDICTION WITH ROOM BALLET

In this section, we use the Room Ballet environment, inspired by the Ballet task described in (Lampinen et al., 2021). Figure 3 illustrates the environment, which consists of multiple rooms. By default, there are 9 rooms, but this can be expanded to 16 and 25 in Exp-2. Each room represents a "place" and contains a dancer. In each episode, the agent is placed randomly in one of the rooms and moves through the rooms via random walk. Upon entering a room, the agent observes a complete dance performance consisting of 32 frames, performed by the dancer in that room. In each episode, a *dancer type* (i.e., the appearance of a dancer) and a *dance type* (i.e., the performance) are randomly and independently assigned to each room.

Figure 3: The Room Ballet environment.

Exp-1. Implicit derivation of spatial information in transformers

To verify that the transformers can derive the spatial information, we designed a spatial reasoning task on the Room Ballet environment, which we call the Next Ballet task. In this task, the agent visits 40 rooms (1280 steps) through a random walk with the four actions. Afterward, when provided with a query image showing a dancer's appearance, it is tasked to predict the type of dance that was performed in *the next room of the queried dancer* in clockwise order.

Although transformers have demonstrated impressive performance across various applications (Brown et al., 2020; Dosovitskiy et al., 2020; He et al., 2022; Parisotto et al., 2020; Micheli et al., 2022), it is unclear whether they can infer spatial information from navigation trajectories consisting of observations, actions, and time. To investigate this, we evaluate the following models: a standard transformer only with temporal order information T-FIFO, a transformer with temporal order and action T-FIFO+A, and a transformer with spatial information SAT-FIFO. Also, to ensure that memory capacity does not affect the results, we set the memory size to be large enough to store all observations.

As shown in the Figure 4 (a), we can see that the transformer using only temporal order without actions (T-FIFO) completely fails on this spatial reasoning task, suggesting its inability to develop an internal representation of the space from the ordered sequence of observations, as expected in the Thought Experiment. When we additionally provide the action information (T-FIFO+A), it began to show a better performance but still significantly inadequate to accurately solve the spatial reasoning problem by inferring the dancer's room location. These results suggest that inferring spatial information from observations and actions is not trivial in practice, though it may not be totally impossible. On the contrary, explicitly incorporating spatial information, SAT successfully solves the task, demonstrating the importance of having explicit spatial information in spatial reasoning tasks.

Exp-2. Place-centric hierarchical reading vs Time-centric hierarchical reading

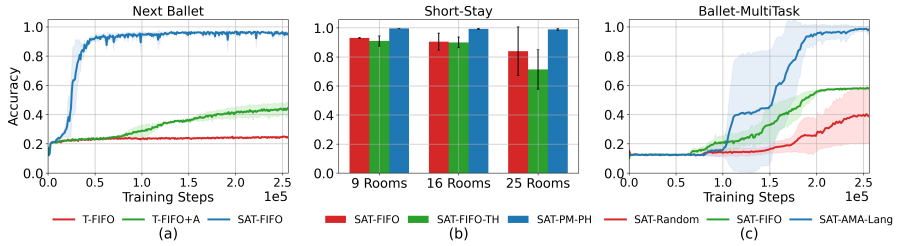


Figure 4: (a) The performance comparison of transformer memories equipped with different embeddings on the spatial reasoning task. (b) The performance comparison of unstructured memory and temporally/spatially structured memory on a spatial reasoning task in a large environment. (c) The performance comparison of the AMA model and predefined memory allocation strategies on a spatial reasoning task.

To demonstrate that place-centric chunking proposed in Section 2.2, is more beneficial for spatial task compared to traditional time-based chunking, we introduce a new and more realistic task called "Short Stay" in the Ballet environment. In this task, unlike the previous one, the agent is not required to stay in a room until it observes the entire dance. Instead, the agent exhibits more natural behavior by being able to move to a neighboring room at any time step. After making 1280 such moves, the agent is presented with a query dancer image and tasked with predicting its dance type. To verify the efficacy of the place-centric hierarchical SAT, we compared three models: (i) SAT-FIFO, (ii) SAT-FIFO with time-centric hierarchical read (SAT-FIFO-TH), and (iii) SAT-PM with place-centric hierarchical read (SAT-PM-PH). The accuracies are measured after 200K training steps, and the chunk size is set to 32. More details about this task are provided in Appendix C.3.

As shown in Figure 4 (b), it is evident that SAT-PM-PH consistently outperforms SAT-FIFO-TH and SAT-FIFO in all scenarios. Furthermore, as the number of rooms increases, the performance gap between SAT-FIFO-TH and SAT-PM-PH also widens. We observe that this is because SAT-FIFO-TH allows for temporally consecutive experiences from different places to be grouped together within a chunk. Similarly, experiences from the same place can be spread across multiple chunks. Consequently, gathering information about an event that occurred in a specific location by collecting scattered experience frames across multiple chunks remains a more challenging task for SAT-FIFO-TH, despite its use of place encoding and computational speed enhancement due to the hierarchy. On the other hand, SAT-PM-PH easily overcomes this challenge since a chunk contains only experiences from the same place. When compared to the non-hierarchical SAT-FIFO, SAT-PM-PH not only achieves better accuracy but also demonstrates faster inference speed, as in Table 1 in Appendix. This is because SAT-PM-PH only attends to the memory in the top-k relevant chunks, while SAT-FIFO needs to attend to the entire memory.

Exp-3. Learning to select memory allocation strategy with AMA

In this section, we assume that the memory capacity is limited and demonstrate that with SAT-AMA, it is possible to learn to find a proper memory management strategy that best fits the downstream task and that it is better than always using FIFO. We first introduce four strategies: First-In-First-Out (FIFO), Last-In-First-Out (LIFO), Most-Visited-First-Out (MVFO), and Least-Visited-First-Out (LVFO). FIFO/LIFO replace the oldest/newest observation in the memory. MVFO/LVFO replace the oldest memory from the most/least frequently visited room. Note that whether these strategies are good or not is not the primary value of this work. Rather, they are examples of prior strategies that can be designed.

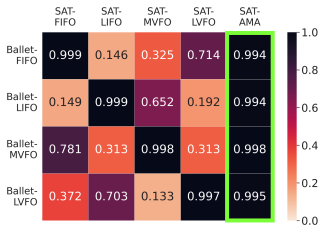


Figure 5: The accuracy of different task-strategy pairs.

Then, we introduce four different tasks, one for each strategy: Ballet-FIFO, Ballet-LIFO, Ballet-MVFO, and Ballet-LVFO. In the tasks, which consist of a total of 9 rooms as shown in Figure 3, the agent changes rooms 17 times (for a total of 576 steps) using random walk. This allows the agent to visit a room multiple times, potentially encountering multiple distinct dancers in a room. It is important to note that we limit the memory capacity to 288 and thus a half of the total episode steps must be chosen to be removed according to the management strategy. We note that the agent changes rooms after observing a complete dance performance in those tasks.

In each task, the agent is asked to predict the dance type of a queried dancer image. The selection of the query dancer follows a specific strategy, ensuring that it can only be answered correctly if the

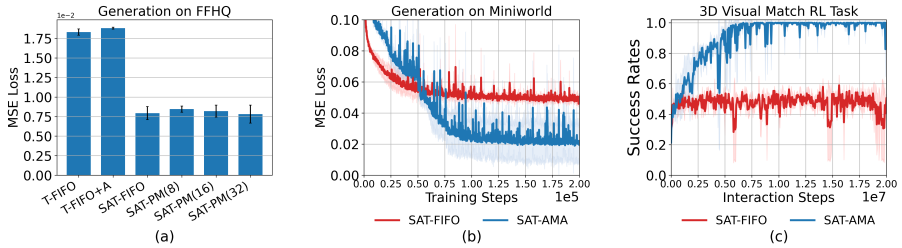


Figure 6: (a) The generation performance comparison for the navigation task on facial images from the FFHQ dataset. For SAT-PM, the numbers in parentheses are the number of the approximated places. (b) The generation performance comparison on the MiniWorld environment, which is a 3D non-grid environment. (c) The performance on the Visual Match task on the MiniWorld environment.

correct storage strategy was used. For instance, in the Ballet-LIFO task, the query dancer is always selected from the oldest half of the episode. Therefore, if the FIFO strategy was used, the memory does not contain the oldest half, making it impossible to correctly answer the query. However, if LIFO is chosen, the memory always retains the necessary experience to answer the query. Similarly, in the Ballet-MVFO/LVFO tasks, the agent is asked to predict the most recently observed dances within each room and the dances in the most frequently visited room, respectively. Importantly, given a task, the agent does not know what the best strategy is but needs to learn to choose it via AMA. Note that we use place-centric hierarchical read for selecting MVFO/LVFO, and time-centric hierarchical read for selecting in FIFO/LIFO in AMA. More details about the tasks are provided in Appendix C.4.

Figure 5 depicts the performance of different task-strategy pairs. As anticipated, we observe strong performance in the diagonal terms of the matrix when the strategy aligns with the task. Conversely, when the strategy does not match (i.e., the off-diagonal terms), the performance deteriorates. Notably, in the 5th column, we present the performance of SAT-AMA. SAT-AMA achieves comparable performance to strategies specifically designed for each task (the diagonal terms) as it learns to discover suitable strategies for each task.

Exp-4. Generalization of SAT-AMA to multi-task descriptions

In Exp-3, we demonstrated that SAT-AMA can learn to select an appropriate strategy for a single task. However, in this specific scenario, it is possible to find the correct strategy by trying out all configurations, as there are only four tasks and four strategies. In a more realistic scenario, it is common to ask the agent to solve multiple tasks with diverse formats, such as language descriptions (as described in Sec. 2.3). In such cases, the model should be able to generalize to the diverse variations of task descriptions. In this setting, it is impossible to try all possible configurations.

To test this, we mapped each task in Exp-3 to a set of natural language descriptions. We created 20 different descriptions for each task. For example, for the Ballet-LVFO task, we provided descriptions such as "*Recall the dancers in the room the agent frequented the most*" or "*Reflect on the dancers in the room that seemed to be the agent's favorite*". During training, we randomly selected 64 mixed descriptions (16 per task) and used the remaining 16 descriptions (4 per task) to evaluate SAT with AMA. We name this version of SAT as SAT-AMA-Lang. All descriptions used in the experiment can be found in Appendix C.5. For AMA, the description is given as an embedding from the OpenAI embeddings API [2]. The results are shown in Figure 4 (c). It is important to note that the accuracy in the evaluation is an average of three different training/evaluation description sets. In this case, SAT-AMA achieved an accuracy of over 95%. This indicates that AMA can be extended to generalize to the diverse variations of task descriptions and has potential in broad applications such as language-instructed robots.

3.2 Episodic Imagination via Image Generation

In this section, we demonstrate the potential of applying SAT to more complex tasks and environments: action-conditioned episodic image generation requiring multi-step spatial reasoning, and episodic image generation with AMA.

Exp-5. Action-conditioned generation in the FFHQ world

[2]We took the embedding from the model text-embedding-ada-002. https://platform.openai.com/docs/guides/embeddings/what-are-embeddings

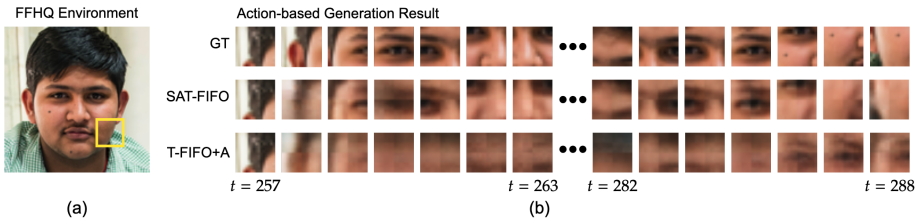


Figure 7: (a) FFHQ environment. The agent observes the partial view (yellow box) in the face image from FFHQ dataset. (b) Action-based generation result of SAT-FIFO and T-FIFO+A after Observation phase.

We demonstrate the capability of the SAT to generate conditioning on actions while navigating an environment to see its potential to implement an episodic world model (Gregor et al., 2019; Hafner et al., 2020; Micheli et al., 2022; Chen et al., 2022). To test this, we designed a navigation task on the facial images from FFHQ dataset (Karras et al., 2019), where 62,000 face images are used for training and 7,000 images among the remaining images are used for evaluation. The environment is a face image which is considered as a 10×10 grid map. The agent observes a partial image of the face, as shown in Figure 7 (a), while navigating the image provided with an action (up, down, left, right) at each step. During the observation phase, the agent spawns in a random location and performs a random walk for 256 steps. During the generation phase, at a random position, it is given a sequence of random actions of length 32. Then, the task is to generate images corresponding to the queried action sequence by utilizing the episodic memory constructed during the observation phase. Therefore, this task requires multi-step spatial reasoning. See Figure 17 in Appendix for more visualization. For this task, 2-D sinusoidal positional embedding that encodes x and y axes separately, is utilized for the space embedding. The comparative analysis of the embedding types for the space embedding is discussed in Appendix D.9.

To verify the efficacy of the SAT-FIFO, we compared it with T-FIFO and T-FIFO+A. The results are shown in Figure 6 (a) and 7 (b). Since this task is a harder version of the Next Ballet task in Exp-1 due to a larger map, T-FIFO and T-FIFO+A failed to solve it. SAT-FIFO, on the other hand, outperformed them and generated high-quality images. We note that this result is from the evaluation set which contains facial images that are unseen during training. This result suggests that SAT can handle complex spatial reasoning on unseen visual environments. Additionally, we investigated the effect of using approximate place structures by changing the number of clusters ($N = 8, 16, 32$). As shown in 6 (a), SAT-PM(N) showed robust performance across different place clustering. See Appendix D.7 for more results and discussion.

Exp-6. Image generation in 3D first-person view environment

Next, we evaluate SAT with AMA for the image generation in the MiniWorld environment (Chevalier-Boisvert, 2018), a 3D environment where an agent navigates in the first-person view setting. As shown in Figure 11 in the Appendix, the agent is placed at the center of a room with four dancers positioned at each side of the room (North, South, East, and West), each performing a distinct dance. The agent observes the room by rotating at a random degree between 3 and 7 in a clockwise order. Afterwards, the agent is tasked with generating a dance video featuring the queried dancer. Due to the rotation, during the observation phase, the agent observes the dance from various viewpoints, as shown in Figure 11. However, during the generation phase, the agent is asked to generate the dance from the front view. Therefore, the model cannot simply copy the dance from its episodic memory, but instead needs to develop a proper representation of the dance from the episodic memory and decode it accordingly. Similar to the Ballet tasks, we tested four different types of queries: FIFO, LIFO, MVFO, and LVFO. As depicted in Figures 6 (b) and 11, we can see that SAT-AMA successfully learns to choose the appropriate strategy for the given task. In contrast, SAT-FIFO performs worse because it loses necessary memory if it is older than the memory capacity. More details about the experiment are provided in Appendix C.6.

3.3 Reinforcement Learning Agent

Lastly, we test an RL agent equipped with SAT-AMA memory. The task involves two rooms. The agent is initially placed in a room with a single box of a random color (either blue or green). After staying four steps in the room, the agent is transported to the second room which is larger and contains multiple yellow boxes, and navigates for 80 steps.

After this observation phase, the agent is warped back to the first room, where it is tasked to collect the box that matches the color of the box observed initially. The memory capacity is limited to 40, a half of the number of total steps in the second room. We use Proximal Policy Optimization (PPO) (Schulman et al., 2017) for agent learning. AMA is designed to learn through the reward from the environment and to choose a strategy between FIFO and MVFO. As shown in Figure 6 (c), SAT-AMA successfully learned to select the appropriate strategy (MVFO) and solve the task. While it only needs to choose one out of two strategies, it is important to note that this still a challenging task where AMA needs to learn through sparse rewards and many interaction steps, and in the first-person view 3D environment.

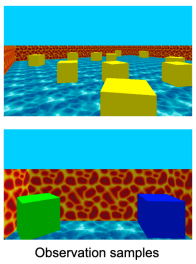


Figure 8: RL environment visualization.

4 Related Works

Transformers have been widely used in various applications (Brown et al., 2020; He et al., 2022; Parisotto et al., 2020; Chen et al., 2021), however, their high computational cost has been a major drawback, especially for reinforcement learning (Parisotto & Salakhutdinov, 2021). To address this issue, several methods have been proposed to reduce the computational cost of transformers without sacrificing their long-term memorization ability (Parisotto & Salakhutdinov, 2021; Lampinen et al., 2021). In this paper, we propose a new type of hierarchical read operation that is more efficient for place-centric tasks. We also propose a memory management method based on reinforcement learning that aims to optimize memory usage.

In parallel, there has been growing interest in using spatial information for embodied agents (Zhang et al., 2017; Rosenbaum et al., 2018; Eslami et al., 2018; Singh et al., 2019; Yoon et al., 2020). Some studies have focused on using spatial information to improve navigation performance (Zhang et al., 2017; Rosenbaum et al., 2018), while others have aimed to construct cognitive maps (Eslami et al., 2018; Singh et al., 2019; Yoon et al., 2020). In this study, we investigate incorporating spatial information into episodic memory. We show that this can improve performance and memory usage efficiency on spatial reasoning tasks without sacrificing performance.

5 Conclusion and Limitations

Transformer-based episodic memory has utilized temporal order to serialize experience frames. In this paper, we explore the potential of incorporating the spatial axis as another fundamental aspect of the physical world. We argue that this is crucial for embodied agents and introduce the concept of Spatially-Aware Transformers (SAT). We propose different SAT architectures, starting from the simplest SAT-FIFO to SAT with place memory, which includes a hierarchical episodic memory centered around places. Furthermore, we introduce the Adaptive Memory Allocation (AMA) method, which provides a more flexible memory management strategy beyond the FIFO memory writing approach. Through experiments, we assess the improved performance of these methods and demonstrate their applicability to a wide range of machine learning problems.

Limitations. Despite the aforementioned contributions, this work is based on several simplifying assumptions as the first to propose the concept of spatially-aware transformers. One of the main assumptions is its reliance on spatial annotation. Although spatial annotation is widely available in many embodied agent applications, incorporating the learning of spatial representation alongside its usage could enhance the applicability of the proposed models. Another limitation is the generality of the AMA method. While it offers more flexibility than FIFO, it still relies on a set of predefined strategies. While it would be ideal to discover the strategy from scratch, previous works like the Neural Turing Machine (Graves et al., 2014; 2016) have demonstrated that this approach can introduce significant training complexity, limiting its practical application. Thus, there is a need for a new method that improves both flexibility and ease of training. Lastly, while in this work we focused on spatial reasoning tasks, there would also be non-spatial reasoning tasks in spatial environments. Generalizing the proposed method to handle such cases would also be an interesting future direction.

6 ETHICS STATEMENT

We state some potential negative societal impacts of this work. Although we do not have any concerns in its current form, it could be later connected to a general-purpose autonomous agent that could be used for malicious purposes, or that could violate people's privacy. Another potential problem may come from our model's assumption, which may require place information in public places for real-world applications in the future. This could cause privacy issues.

7 REPRODUCIBILITY STATEMENT

To facilitate the reproducibility of our work, we have provided the pseudo codes, model architecture and hyperparameters in Appendix B.3, and B.4. We will also release the source code for our models and experiments.

ACKNOWLEDGEMENT

This work is supported by Brain Pool Plus (BP+) Program (No. 2021H1D3A2A03103645) through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT. JY thanks to SAP for their support.

REFERENCES

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

György Buzsáki and David Tingley. Space and time: the hippocampus as a sequence generator. *Trends in cognitive sciences*, 22(10):853–869, 2018.

Chang Chen, Yi-Fu Wu, Jaesik Yoon, and Sungjin Ahn. Transdreamer: Reinforcement learning with transformer world models. *arXiv preprint arXiv:2202.09481*, 2022.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021.

Maxime Chevalier-Boisvert. Miniworld: Minimalistic 3d environment for rl & robotics research. `https://github.com/maximecb/gym-miniworld`, 2018.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

Peter J Denning. The locality principle. *Communications of the ACM*, 48(7):19–24, 2005.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Arne D Ekstrom and Charan Ranganath. Space, time, and episodic memory: The hippocampus is all over the cognitive map. *Hippocampus*, 28(9):680–687, 2018.

Ahmed El-Rabbany. *Introduction to GPS: the global positioning system*. Artech house, 2002.

S. M. Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S. Morcos, Marta Garnelo, Avraham Ruderman, Andrei A. Rusu, Ivo Danihelka, Karol Gregor, David P. Reichert, Lars Buesing, Theophane Weber, Oriol Vinyals, Dan Rosenbaum, Neil Rabinowitz, Helen King, Chloe Hillier, Matt Botvinick, Daan Wierstra, Koray Kavukcuoglu, and Demis Hassabis. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018. doi: 10.1126/science.aar6170. URL https://www.science.org/doi/abs/10.1126/science.aar6170.

Ramsey Faragher and Robert Harle. Location fingerprinting with bluetooth low energy beacons. *IEEE journal on Selected Areas in Communications*, 33(11):2418–2428, 2015.

Marco Fraccaro, Danilo Rezende, Yori Zwols, Alexander Pritzel, SM Ali Eslami, and Fabio Viola. Generative temporal models with spatial memory for partially observed environments. In *International conference on machine learning*, pp. 1549–1558. PMLR, 2018.

Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

Alex Graves, Greg Wayne, Malcolm Reynolds, Tim Harley, Ivo Danihelka, Agnieszka Grabska-Barwińska, Sergio Gómez Colmenarejo, Edward Grefenstette, Tiago Ramalho, John Agapiou, et al. Hybrid computing using a neural network with dynamic external memory. *Nature*, 538(7626): 471–476, 2016.

Karol Gregor, Danilo Jimenez Rezende, Frederic Besse, Yan Wu, Hamza Merzic, and Aaron van den Oord. Shaping belief states with generative environment models for rl. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/2c048d74b3410237704eb7f93a10c9d7-Paper.pdf.

William H Guss, Brandon Houghton, Nicholay Topin, Phillip Wang, Cayden Codel, Manuela Veloso, and Ruslan Salakhutdinov. Minerl: A large-scale dataset of minecraft demonstrations. *arXiv preprint arXiv:1907.13440*, 2019.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

Chia-Chun Hung, Timothy Lillicrap, Josh Abramson, Yan Wu, Mehdi Mirza, Federico Carnevale, Arun Ahuja, and Greg Wayne. Optimizing agent behavior over long time scales by transporting value. *Nature communications*, 10(1):5223, 2019.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Andrew Lampinen, Stephanie Chan, Andrea Banino, and Felix Hill. Towards mental time travel: a hierarchical memory for reinforcement learning agents. *Advances in Neural Information Processing Systems*, 34:28182–28195, 2021.

Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, pp. 3. Atlanta, Georgia, USA, 2013.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample efficient world models. *arXiv preprint arXiv:2209.00588*, 2022.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

Emilio Parisotto and Ruslan Salakhutdinov. Efficient transformers in reinforcement learning using actor-learner distillation. *arXiv preprint arXiv:2104.01655*, 2021.

Emilio Parisotto, Francis Song, Jack Rae, Razvan Pascanu, Caglar Gulcehre, Siddhant Jayakumar, Max Jaderberg, Raphael Lopez Kaufman, Aidan Clark, Seb Noury, et al. Stabilizing transformers for reinforcement learning. In *International conference on machine learning*, pp. 7487–7498. PMLR, 2020.

Thanujeni Pathman, Christine Coughlin, and Simona Ghetti. Space and time in episodic memory: Effects of linearity and directionality on memory for spatial location and temporal order in children and adults. *PLoS One*, 13(11):e0206999, 2018.

Denis Perrin and Kourken Michaelian. Memory as mental time travel. *The Routledge handbook of philosophy of memory*, pp. 228–239, 2017.

Dan Rosenbaum, Frederic Besse, Fabio Viola, Danilo J Rezende, and SM Eslami. Learning models for visual 3d localization with implicit mapping. *arXiv preprint arXiv:1807.03149*, 2018.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Daniel L Schacter, Roland G Benoit, and Karl K Szpunar. Episodic future thinking: Mechanisms and functions. *Current opinion in behavioral sciences*, 17:41–50, 2017.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Gautam Singh, Jaesik Yoon, Youngsung Son, and Sungjin Ahn. Sequential neural processes. *Advances in Neural Information Processing Systems*, 32, 2019.

Thomas Suddendorf, Donna Rose Addis, and Michael C Corballis. Mental time travel and the shaping of the human mind. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364 (1521):1317–1324, 2009.

Takafumi Taketomi, Hideaki Uchiyama, and Sei Ikeda. Visual slam algorithms: A survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9(1):1–11, 2017.

Benigno Uria, Borja Ibarz, Andrea Banino, Vinicius Zambaldi, Dharshan Kumaran, Demis Hassabis, Caswell Barry, and Charles Blundell. The spatial memory pipeline: a model of egocentric to allocentric understanding in mammalian brains. *bioRxiv*, 2020. doi: 10.1101/2020.11.11.378141. URL https://www.biorxiv.org/content/early/2020/11/12/2020.11.11.378141.

Marieke van Asselen, Rob Van der Lubbe, and Albert Postma. Are space and time automatically integrated in episodic memory? *Memory*, 14(2):232–240, 2006.

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 C. R. Whittington, Joseph Warren, and Timothy E. J. Behrens. Relating transformers to models and neural representations of the hippocampal formation. 2022.

Jaesik Yoon, Gautam Singh, and Sungjin Ahn. Robustifying sequential neural processes. *International Conference on Machine Learning*, 2020.

Jingwei Zhang, Lei Tai, Ming Liu, Joschka Boedecker, and Wolfram Burgard. Neural slam: Learning to explore with external memory. *arXiv preprint arXiv:1706.09520*, 2017.