RE:FRAME – RETRIEVING EXPERIENCE FROM ASSOCIATIVE MEMORY

Daniil Zelezetsky[1] **Egor Cherepanov**[1,2] **Alexey K. Kovalev**[1,2] **Aleksandr I. Panov**[1,2]
[1]MIPT, Dolgoprudny, Russia [2]AIRI, Moscow, Russia
zelezetskii.dv@phystech.edu, {cherepanov,kovalev,panov}@airi.net

ABSTRACT

Transformers have demonstrated strong performance in offline reinforcement learning (RL) for Markovian tasks, due to their ability to process historical information efficiently. However, in partially observable environments, where agents must rely on past experiences to make decisions in the present, transformers are limited by their fixed context window and struggle to capture long-term dependencies. Extending this window indefinitely is not feasible due to the quadratic complexity of the attention mechanism. This limitation led us to explore other memory handling approaches. In neurobiology, associative memory allows the brain to link different stimuli by activating neurons simultaneously, creating associations between experiences that occurred around the same time. Motivated by this concept, we introduce **Re:Frame** (**R**etrieving **E**xperience **Fr**om **A**ssociative **Me**mory), a novel RL algorithm that enables agents to better utilize their past experiences. Re:Frame incorporates a long-term memory mechanism that enhances decision-making in complex tasks by integrating past and present information.

1 INTRODUCTION

Memory is fundamental to human intelligence, enabling us to accumulate experiences, learn from past interactions, and make informed decisions in complex environments (Tulving, 2002; Squire, 2004; Baddeley, 2010). This cognitive capability allows humans to seamlessly integrate historical information with present observations, facilitating adaptive behavior across diverse scenarios (Eichenbaum, 2017; Parr et al., 2020; 2022).

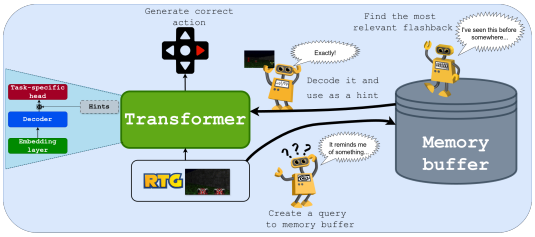

Figure 1: Re:Frame – an associative memory framework that enables retrieval of relevant experiences for RL agents.

In the field of reinforcement learning (RL), significant advances have led to agents that can surpass human performance (Mnih et al., 2015; Silver et al., 2017) in Markovian tasks – environments where optimal decision-making depends solely on the current state (Sutton & Barto, 2018). However, real-world scenarios frequently present partial observability challenges, where complete environmental information is not immediately available (Kaelbling et al., 1996). For instance, while humans can effortlessly recall the location of an object days after placing it, traditional RL agents struggle with such memory-intensive tasks without specialized memory mechanisms (Wayne et al., 2018; Parisotto et al., 2020).

Among various approaches to embedding memory in artificial agents (Zaremba & Sutskever, 2015; Oh et al., 2016; Wayne et al., 2018), transformer architectures (Vaswani et al., 2017) have emerged as a promising solution, largely due to their remarkable success in processing sequential data, particularly in natural language tasks (Achiam et al., 2023; Guo et al., 2025). However, these transformer-based agents face a critical limitation in partially observable RL environments: their effectiveness dramatically diminishes when crucial information extends beyond their fixed context window. This constraint restricts their ability to maintain and utilize long-term memories effectively.

To address these limitations, we draw inspiration from human associative memory – a fundamental cognitive mechanism that enables the brain to form connections between different stimuli and

experiences (Polson, 1975; Steinberg & Sompolinsky, 2022). When humans encounter a situation, they naturally recall and utilize relevant past experiences through associative links, even if these experiences occurred in the distant past (Schacter, 1999). This biological process allows for efficient retrieval and application of pertinent information without the need to sequentially process all intervening experiences.

Motivated by this cognitive mechanism, we introduce **Re:Frame** – **R**etrieving **E**xperience **Fr**om **A**ssociative **Me**mory – a novel method for RL agents inspired by the associative memory principles of the human brain. Our approach enables agents to form and utilize associative connections between current observations and relevant past experiences, effectively bypassing the context window limitations of traditional transformer-based architectures. Re:Frame creates a memory space where experiences are encoded and organized in a way that facilitates rapid retrieval of relevant information based on contextual similarities, rather than temporal proximity.

The key innovation of Re:Frame lies in its ability to dynamically access and utilize relevant historical information through associative retrieval, regardless of when that information was originally encountered. This mechanism allows our agents to maintain effective decision-making capabilities in partially observable environments, even when crucial information lies far beyond the traditional context window. By combining the sequential processing capabilities of transformers with an associative memory mechanism, Re:Frame achieves robust performance in memory-intensive tasks.

Our contribution can be summarized as follows:

- We introduce **Re:Frame**, a novel associative memory framework for RL that enables the retrieval of relevant experiences independent of temporal distance, thus addressing the fundamental challenge of long-term information retention in memory-intensive tasks.
- We demonstrate that Re:Frame can be effectively integrated with existing RL architectures, significantly improving their performance on partially observable tasks through efficient memory utilization and retrieval mechanisms.

2 Related Works

Associative memory mechanisms, inspired by biological neural systems, have been explored in various machine learning contexts. Hopfield networks (Hopfield, 1982) represent an early attempt to implement associative memory in artificial neural networks. Neural Turing Machines (NTM) (Graves, 2014) introduced external memory with both content-based and location-based addressing, while Associative Recurrent Memory Transformer (ARMT) (Rodkin et al., 2024) extended transformer architecture with an associative memory to enhance its ability to handle long-term dependencies.

The principles of associative memory have also been investigated in RL. Thus, Associative Memory Prioritized Experience Replay (AMPER) (Li et al., 2022) utilizes associative memory to accelerate prioritized experience replay in deep RL. Fast Weight Memory (FWM) (Schlag et al., 2020) enhances Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) architecture with associative memory capabilities, enabling efficient meta-learning and associative inference in RL. Self-attentive Associative Memory (SAM) (Le et al., 2020) implements a dual-memory system that combines item storage with relational memory, enabling both memorization and relational reasoning capabilities in RL tasks. Associative Search Network (ASN) (Barto et al., 1981) introduces a self-learning associative memory that optimizes output patterns based on reinforcement signals, enabling autonomous learning of sensory-motor control without explicit supervision. Episodic Reinforcement Learning with Associative Memory (ERLAM) (Zhu et al., 2020) improves sample efficiency in RL by building a graph-based associative memory that connects related experiences and enables rapid value propagation through reverse-trajectory updates.

While these approaches demonstrate the potential of associative memory in RL, they focus primarily on specific architectural modifications. In contrast, Re:Frame can potentially be integrated into any RL agent architecture without modifying its core structure. Our approach differs by using a dedicated associative memory buffer that stores and retrieves experiences based on their similarity, rather than relying on temporal relationships or explicit graph structures. This design allows for more flexible and context-aware memory retrieval, which is particularly beneficial for tasks requiring long-term memory retention.

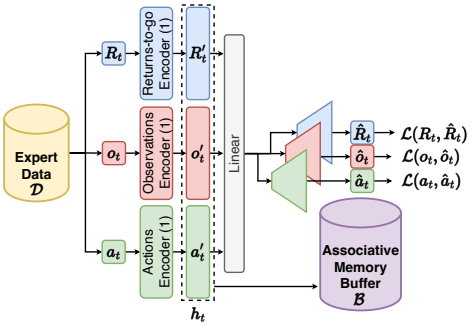

Figure 2: The process of Associative Memory Buffer generation using AE on expert data.

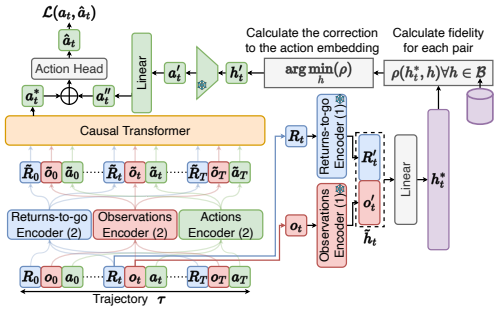

Figure 3: Integration of Re:Frame with DT to support decision-making process.

3 Background

Partially Observable Markov Decision Process. A Partially Observable Markov Decision Process (POMDP) extends the standard Markov Decision Process (MDP) framework to scenarios where agents cannot directly observe the complete state of the environment (Kaelbling et al., 1998). Formally, a POMDP is defined as a tuple (S, A, T, R, Ω, O), where S is the state space, A is the action space, T is the transition function, R is the reward function, Ω is the observation space, and O is the observation function. At each timestep, instead of observing the true state s_t, the agent receives an observation $o_t \in \Omega$ that may only partially reflect the underlying state.

Algorithm 1 Re:Frame-DT Integration

Require: AMB $\mathcal{B} \in \mathbb{R}^{T \times N}$, trajectory τ
Ensure: Predicted actions $\hat{a}$
Memory Retrieval:
for $(R_t, o_t, a_t) \in \tau$ **do**
  $R'_t \leftarrow \text{RtgEnc1}(R_t)$, $o'_t \leftarrow \text{ObsEnc1}(o_t)$
  $h^*_t \leftarrow \text{Linear}(\text{concat}(R'_t, o'_t))$
  $h'_t \leftarrow \arg\min_{h \in \mathcal{B}} \|h^*_t - h\|_2^2$
  $a'_t \leftarrow \text{ActDec1}(h'_t)$
  $a''_t \leftarrow \text{Linear}(a'_t) \in \mathbb{R}^{1 \times D}$
end for
Action Generation:
$\tau' \leftarrow$ Embed Sequence(τ)
$a^* \leftarrow \text{Transformer}(\tau') \in \mathbb{R}^{T \times D}$
$\hat{a} \leftarrow \text{ActHead}(a^* + a'') \in \mathbb{R}^{T \times D}$
return $\hat{a}$ with loss $\mathcal{L}(a, \hat{a})$

Offline Reinforcement Learning. Offline RL learns policies from a fixed dataset $\mathcal{D}$ without environment interaction. Each trajectory $\tau \in \mathcal{D}$ consists of triplets (r_t, o_t, a_t), containing immediate reward r_t, observation o_t, and action a_t. Decision Transformer (DT) (Chen et al., 2021) reformulates RL as a sequence modeling problem by introducing return-to-go $R_t = \sum_{k=t}^{T} r_k$, which represents the cumulative future rewards from timestep t. DT processes sequences of (R_t, o_t, a_t) tokens using a transformer architecture to autoregressively predict actions that achieve the desired return. By conditioning on different target returns during inference, DT can generate behaviors of varying quality from the same trained model. We selected DT as our base architecture as its attention-based nature provides a clear way for measuring the impact of Re:Frame's memory enhancement capabilities on long-term information retention tasks.

Autoencoder. An Autoencoder (AE) (Rumelhart et al., 1986) is a neural network that learns compact data representations through an encoder-decoder architecture. The encoder $f_\theta : \mathcal{X} \to \mathcal{Z}$ maps input data to a lower-dimensional latent space, while the decoder $g_\phi : \mathcal{Z} \to \mathcal{X}$ reconstructs the original input. Training minimizes the reconstruction loss $\mathcal{L}(\theta, \phi) = \|x - g_\phi(f_\theta(x))\|^2$. We leverage AE to create efficient encodings of agent experiences for memory-intensive decision-making.

4 Re:Frame Method

The proposed Re:Frame method employs a two-stage training strategy: initially, we train an Autoencoder (AE) to construct the Associative Memory Buffer (AMB), a compressed repository of expert experiences (Figure 2). The AE's parameters are then fixed, ensuring stable memory representations throughout the subsequent learning process. Then, the proposed decision-making framework that leverages this stored information from AMB to enhance the agent's performance (Figure 3).

4.1 Associative Memory Buffer

Our method employs AMB that compresses and stores key environmental events, enabling efficient retrieval of past experiences for decision-making in memory-intensive tasks (see Figure 2).

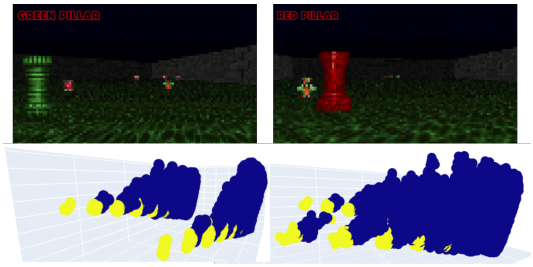

Figure 4: Agent's observation in ViZDoom-Two-Colors (top) and corresponding feature spaces (bottom), where the yellow color indicates the first 45 steps with a visible pillar. Bottom figures demonstrate feature space that was obtained by reducing latent representations into 3-dimensional vectors using PCA algorithm.

The buffer construction process begins by sampling triplets (R_t, o_t, a_t) at timesteps t from the expert trajectory dataset $\mathcal{D}$. Each component of these triplets is processed through dedicated encoders to produce corresponding embeddings (R'_t, o'_t, a'_t). These embeddings are then concatenated into a unified hidden state $h_t = \text{concat}(R'_t, o'_t, a'_t)$ and stored in the Associative Memory Buffer $\mathcal{B}$. A linear transformation maps h_t to a compact latent representation, which is subsequently decoded back into its original components $(\hat{R}_t, \hat{o}_t, \hat{a}_t)$ through separate decoders. During training, we optimize each decoder's reconstruction loss independently using separate optimizers for each component.

4.2 Decision-making with Re:Frame

To evaluate Re:Frame's effectiveness in memory-intensive tasks, we integrated it with Decision Transformer (DT) architecture. As DT lacks mechanisms for processing information beyond its context window, comparing DT and Re:Frame-DT performance directly demonstrates the benefits of our approach. The Re:Frame-DT integration, illustrated in Figure 3 and detailed in Algorithm 1, operates in two stages: Memory Retrieval and Action Generation.

Memory Retrieval. The first stage processes each timestep t of trajectory τ by encoding returns-to-go R_t and observations o_t through pre-trained AE encoders. These embeddings are concatenated ($\tilde{h}_t = \text{concat}(R'_t, o'_t)$) and transformed through a linear layer to match AMB's latent space, producing h^*_t. The AMB $\mathcal{B}$ contains expert demonstrations encoded as latent vectors h, while h^*_t represents the current state. To leverage relevant past experiences, we retrieve h'_t from $\mathcal{B}$ that minimizes $\|h^*_t - h\|_2^2$. This retrieved memory is processed through frozen AE decoder to obtain a'_t, which is further refined through the linear layer to produce the correction vector a''_t.

Action Generation. The second stage processes the input trajectory τ through dedicated encoders to create embedded sequence τ'. This sequence feeds into the transformer to generate action embeddings a^*. The final action is produced by combining a^* with the memory-derived corrections a''_t through the Action Head layer, effectively incorporating both current context and relevant historical information from AMB $\mathcal{B}$.

5 Experiment Setup and Results

We evaluate Re:Frame in two memory-intensive environments: ViZDoom-Two-Colors (Sorokin et al., 2022) and Minigrid-Memory (Chevalier-Boisvert et al., 2023). In **ViZDoom**, the agent must retain the color of a briefly visible pillar that disappears after 45 steps, and collect same-colored items while navigating a hazardous map. Episodes last up to 2100 timesteps, requiring long-term memory for optimal performance (see Figure 4, top; details in subsection A.1). In **Minigrid-Memory**, the agent must find a visual cue and retain it while moving through a corridor in order to identify the correct goal object, effectively testing both memory and credit assignment capabilities (see subsection A.2). The hyperparameters for the models used in these experiments are provided in the Appendix, Table 3.

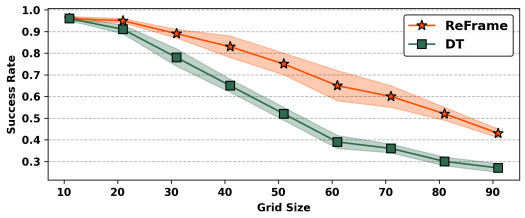

Figure 5: Re:Frame-DT performance in Minigrid-Memory environment.

Table 1: Re:Frame vs baselines in ViZDoom.

Reward[1]	DLSTM	DGRU	DMamba	DT	Re:Frame-DT
R[Total]	13.1 ± 0.6	12.9 ± 0.2	26.9 ± 1.9	24.8 ± 1.4	$\mathbf{34.0} \pm 1.0$
R[Red]	8.8 ± 0.7	9.4 ± 0.5	6.9 ± 0.4	7.2 ± 0.4	$\mathbf{12.4} \pm 1.3$
R[Green]	17.5 ± 1.6	16.3 ± 0.8	46.9 ± 4.2	42.3 ± 3.3	$\mathbf{55.8} \pm 2.1$

Table 2: Effect of buffer size on Re:Frame.

AMB Size	R[Total]	R[Red]	R[Green]
3,000	34.2 ± 0.8	11.5 ± 0.5	56.9 ± 2.0
6,000	33.4 ± 1.4	11.3 ± 0.7	55.4 ± 3.4
9,000	34.0 ± 1.0	12.4 ± 1.3	55.8 ± 2.1

As shown in Figure 5, Re:Frame-DT consistently outperforms the standard DT across all environment sizes in Minigrid-Memory. Notably, both models were trained on environments ranging from 11×11 to 31×31 and evaluated on sizes up to 91×91, demonstrating Re:Frame's superior generalization in out-of-distribution scenarios.

In ViZDoom, we benchmark Re:Frame-DT not only against DT, but also against RNN-based models (DLSTM (Siebenborn et al., 2022), DGRU) and an SSM-based model (DMamba (Gu & Dao, 2023)). We store 9,000 expert triplets in the Associative Memory Buffer (AMB) prior to training and freeze the encoder-decoder weights to ensure stability. Hyperparameter details are provided in Table 3. As summarized in Table 1, Re:Frame-DT outperforms all baselines across total reward and color-specific objectives. Unlike DT, which fails to dominate any metric, Re:Frame-DT delivers substantial performance gains: +72.2% on red, +31.9% on green, and +37.1% overall, indicating that Re:Frame transforms a non-optimal baseline into a leading architecture.

Dependency between buffer size and agent performance. To assess the impact of AMB capacity, we reduce the number of stored expert triplets from 9000 to 6000 and 3000. Results in Table 2 show minimal degradation in performance, suggesting that Re:Frame is robust to reduced memory availability, and full buffer access is not strictly required for strong performance.

6 Limitations and Future Work

While our experiments with DT demonstrate the effectiveness of Re:Frame in memory-intensive environments, several promising directions for future research emerge. First, although Re:Frame is designed to be architecture agnostic, in this paper we only validate its performance with DT, thus validation with other base architectures would provide valuable insights into the framework's versatility. Second, our current evaluation, while promising, focuses on a specific memory-intensive environment. Extending these experiments to a broader range of memory-intensive tasks would help establish the generalizability of the framework. In addition, it would be valuable to verify that Re:Frame maintains performance in classical environments where memory is not essential, to ensure that our approach does not degrade performance in simpler scenarios. Beyond architectural considerations of AMB construction, exploring alternative methods for retrieving experience from the AMB could potentially improve both efficiency and performance, as our current similarity-based mechanism is only one possible approach to leveraging stored experience. Notably, while our current implementation focuses on offline RL, there are no apparent limitations preventing the application of Re:Frame to online RL, suggesting an important direction for future work.

7 Conclusion

In this work, we demonstrated the concept of associative memory mechanism based on latent representations of past experiences. The proposed Re:Frame algorithm can be integrated with any base model, making it a flexible and easy-to-implement tool that does not require additional data for its operation. The idea of storing experience in a compact vector form allows Re:Frame to function without demanding significant computational resources, which is another key advantage. We also experimentally demonstrated the advantages of the Re:Frame algorithm in the memory-intensive VizDoom and Minigrid-Memory environments, significantly improving the performance of the baseline DT architecture.

The primary goal of this work is to introduce the concept rather than to reduce it to strictly defined, architecture-dependent algorithms. We hope that this research will serve as a foundation and inspiration for further studies in this direction.

[1]Results from Cherepanov et al. (2024).

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Alan Baddeley. Working memory. *Current Biology*, 20(4):R136–R140, 2010. ISSN 0960-9822. doi: https://doi.org/10.1016/j.cub.2009.12.014. URL https://www.sciencedirect.com/science/article/pii/S0960982209021332.

Andrew Barto, Richard Sutton, and Peter Brouwer. Associative search network: A reinforcement learning associative memory. *Biological Cybernetics*, 40:201–211, 05 1981. doi: 10.1007/BF00453370.

Edward Beeching, Christian Wolf, Jilles Dibangoye, and Olivier Simonin. Deep reinforcement learning on a budget: 3d control and reasoning without a supercomputer. *CoRR*, abs/1904.01806, 2019. URL http://arxiv.org/abs/1904.01806.

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.

Egor Cherepanov, Alexey Staroverov, Dmitry Yudin, Alexey K. Kovalev, and Aleksandr I. Panov. Recurrent action transformer with memory, 2024. URL https://arxiv.org/abs/2306.09459.

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo Perez-Vicente, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *Advances in Neural Information Processing Systems*, 36:73383–73394, 2023.

Howard Eichenbaum. Memory: organization and control. *Annual review of psychology*, 68(1): 19–45, 2017.

Alex Graves. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8): 1735–1780, nov 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Hung Le, Truyen Tran, and Svetha Venkatesh. Self-attentive associative memory. In *International conference on machine learning*, pp. 5682–5691. PMLR, 2020.

Mengyuan Li, Arman Kazemi, Ann Franchesca Laguna, and X Sharon Hu. Associative memory based experience replay for deep reinforcement learning. In *Proceedings of the 41st IEEE/ACM International Conference on Computer-Aided Design*, pp. 1–9, 2022.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Kirkeby Fidjeland, Georg Ostrovski, Stig Petersen, Charlie Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015. URL https://api.semanticscholar.org/CorpusID:205242740.

Junhyuk Oh, Valliappa Chockalingam, Satinder Singh, and Honglak Lee. Control of memory, active perception, and action in minecraft, 2016. URL https://arxiv.org/abs/1605.09128.

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.

Thomas Parr, Rajeev Vijay Rikhye, Michael M Halassa, and Karl J Friston. Prefrontal computation as active inference. *Cerebral Cortex*, 30(2):682–695, 2020.

Thomas Parr, Giovanni Pezzulo, and Karl J Friston. *Active inference: the free energy principle in mind, brain, and behavior*. MIT Press, 2022.

Peter G. Polson. *The American Journal of Psychology*, 88(1):131–140, 1975. ISSN 00029556. URL http://www.jstor.org/stable/1421672.

Ivan Rodkin, Yuri Kuratov, Aydar Bulatov, and Mikhail Burtsev. Associative recurrent memory transformer. *arXiv preprint arXiv:2407.04841*, 2024.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. 1986. URL https://api.semanticscholar.org/CorpusID:62245742.

Daniel Schacter. The seven sins of memory - insights from psychology and cognitive neuroscience. *The American psychologist*, 54:182–203, 04 1999. doi: 10.1037//0003-066X.54.3.182.

Imanol Schlag, Tsendsuren Munkhdalai, and Jürgen Schmidhuber. Learning associative inference using fast weight memory. *arXiv preprint arXiv:2011.07831*, 2020.

Max Siebenborn, Boris Belousov, Junning Huang, and Jan Peters. How crucial is transformer in decision transformer? *arXiv preprint arXiv:2211.14655*, 2022. URL https://arxiv.org/abs/2211.14655.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy P. Lillicrap, Fan Hui, L. Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550:354–359, 2017. URL https://api.semanticscholar.org/CorpusID:205261034.

Artyom Sorokin, Nazar Buzun, Leonid Pugachev, and Mikhail Burtsev. Explain my surprise: Learning efficient long-term memory by predicting uncertain outcomes. *Advances in Neural Information Processing Systems*, 35:36875–36888, 2022.

Larry R. Squire. Memory systems of the brain: A brief history and current perspective. *Neurobiology of Learning and Memory*, 82(3):171–177, 2004. ISSN 1074-7427. doi: https://doi.org/10.1016/j.nlm.2004.06.005. URL https://www.sciencedirect.com/science/article/pii/S1074742704000735. Multiple Memory Systems.

Julia Steinberg and Haim Sompolinsky. Associative memory of structured knowledge. *Scientific Reports*, 12(1):21808, 2022.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

Endel Tulving. Episodic memory: From mind to brain. *Annual Review of Psychology*, 53(Volume 53, 2002):1–25, 2002. ISSN 1545-2085. doi: https://doi.org/10.1146/annurev.psych.53.100901.135114. URL https://www.annualreviews.org/content/journals/10.1146/annurev.psych.53.100901.135114.

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.

Greg Wayne, Chia-Chun Hung, David Amos, Mehdi Mirza, Arun Ahuja, Agnieszka Grabska-Barwinska, Jack Rae, Piotr Mirowski, Joel Z Leibo, Adam Santoro, et al. Unsupervised predictive memory in a goal-directed agent. *arXiv preprint arXiv:1803.10760*, 2018.

Wojciech Zaremba and Ilya Sutskever. Reinforcement learning neural turing machines-revised. *arXiv preprint arXiv:1505.00521*, 2015.

Guangxiang Zhu, Zichuan Lin, Guangwen Yang, and Chongjie Zhang. Episodic reinforcement learning with associative memory. In *International Conference on Learning Representations*, 2020. URL https://api.semanticscholar.org/CorpusID:212799813.

A Experimental Details

A.1 VizDoom-Two-Colors

We evaluated Re:Frame in ViZDoom-Two-Colors (Sorokin et al., 2022), a memory-intensive environment where an agent must remember a pillar's color (green or red) that disappears after 45 timesteps. The agent navigates an acid floor that depletes health (-10/32 HP per step), collecting matching-colored items for health restoration (+25 HP) and rewards (+1). With episodes spanning 2100 timesteps and a survival bonus of +0.02 per step, success requires long-term retention of the initial color information.

We collected 5000 expert trajectories (90 steps each) using a pre-trained A2C agent (Beeching et al., 2019), achieving an average reward of 4.46. The agent always starts facing the pillar to ensure visual contact before disappearance. Simple color-matching strategies based on recent item collections are ineffective due to occasional mistakes in the training data that can mislead the agent's future decisions.

We ran the training process of Re:Frame on three different seeds and subsequently selected the best checkpoint from each run. Then, each of these three selected checkpoints was evaluated over 50 seeds (25 with green and 25 with red). The resulting rewards were first averaged over the game seeds and then across the three runs, calculating mean reward with std.

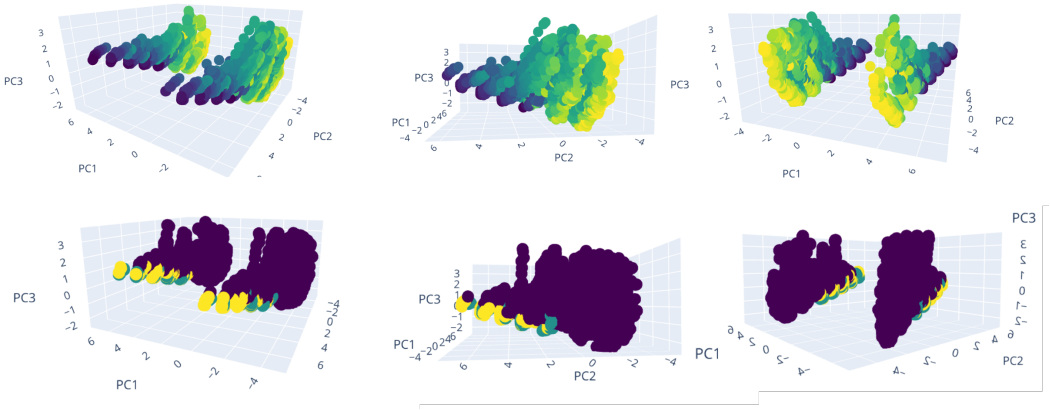

Figure 6: Latent representations of VizDoom triplets colored with the respect to time (top three charts) and with respect to the color of the observable pillar (bottom three charts).

How does the AMB look? Three upper charts on Figure 6 illustrate PCA decomposed latent representations of the expert triplets in the VizDoom environment. The lighter point corresponds to an earlier timestep. Three bottom charts illustrate PCA decomposed triplets but with respect to the color of the observable pillar. Yellow color means that the agent observes the red pillar, green color corresponds to a green pillar. The purple color of the point means that there are no pillars in front of the agent.

A.2 MiniGrid-Memory

Minigrid-Memory (Chevalier-Boisvert et al., 2023) is a grid-based partially observable environment specifically designed to evaluate agents' ability to retain information over long horizons and to test credit assignment capabilities. The environment is structured as a T-shaped maze. At the start of each episode, the agent is placed at a random position within the central corridor. Early in the episode, the agent can access a small room containing a object (a circle or key); this object serves as a cue and must be memorized. Later, upon reaching the end of the corridor, the agent encounters a branching junction and must choose the correct direction—left or right—based on which branch contains an identical object to the one previously seen.

The agent receives a reward that depends on the number of steps from the environment when it turns in the correct direction at a junction. If the agent makes an incorrect choice or exceeds the time limit, the episode terminates with zero reward. Observations are restricted to a 3×3 grid around the agent, reinforcing the need for strong memory mechanisms. To construct the training dataset, we collected 10,000 expert trajectories on maps of size up to 31×31. Expert behavior was generated using the data collection protocol from Cherepanov et al. (2024).

B HYPERPARAMETERS

Table 3 shows the main Re:Frame hyperparameters for experiments in the ViZDoom-Two-Colors and Minigrid-Memory environments. For DT and Re:Frame-DT, we used the same Transformer hyperparameters for a fair comparison.

Table 3: DTand Re:Frame-DT hyperparameters used in ViZDoom-Two-Colors (ViZDoom2C) and Minigrid-Memory environments.

Hyperparameter	**ViZDoom2C**	**Minigrid-Memory**
Number of layers	6	8
Number of attention heads	8	10
Embedding dimension	128	64
Latent memory dim	30	30
Memory buffer size	9000	3000
Context length K	30	30
Hidden dropout	0.2	0.2
Attention dropout	0.05	0.05
Max epochs	100	250
Batch size	50	32
Weight decay	0.1	0.1
Loss function	CE	CE
Optimizer	AdamW	AdamW
Learning rate	3e-4	1e-3
AdamW (β_1, β_2)	(0.9, 0.95)	(0.9, 0.95)