CONTEXTUAL EPISODIC MEMORY FOR CONTEXT-AWARE ROBOT NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Embodied navigation in partially observable environments requires agents to retain information acquired earlier in an episode, since relevant objects, rooms, and landmarks may become inaccessible from the current observation. We propose *Contextual Episodic Memory (CEM)*, a framework that progressively constructs episodic memory by integrating previously accumulated context into newly encountered observations. CEM selectively retains informative visual observations and associates each with the agent's 3D pose. Each new visual-spatial representation attends to stored episodic representations to retrieve relevant context, which is integrated with the current representation before being stored as the next memory entry. This produces an evolving representation of navigation experience that complements the current observation during decision making. We evaluate CEM in the photorealistic environment on two complementary tasks that require episodic memory in distinct ways: multi-room object search, where an agent must use information about previously observed objects and their locations after they are no longer visible, and target-room navigation, where an agent must retain scene-level experience gathered along the trajectory to navigate toward a room from varying starting locations. We compare CEM with recurrent, external-memory, and Transformer-based navigation approaches and analyze the contributions of informative observation selection and contextual memory construction. Results demonstrate that progressively constructed episodic representations improve navigation under partial observability by enabling agents to make decisions using experience accumulated throughout the episode.

1 INTRODUCTION

Embodied navigation requires agents to retain and use information acquired earlier in an episode. In partially observable environments, objects, rooms, landmarks, or visual configurations encountered earlier may no longer be visible from the current viewpoint. A policy based only on the current observation therefore cannot fully exploit past experience. Effective navigation requires memory that accumulates experience while preserving the contextual relationships between observations. Recurrent policies based on LSTMs and GRUs Vennerød et al. (2021); Chung et al. (2014) address this challenge by compressing past observations into a hidden state, while explicit memory architectures such as Differentiable Neural Computers (DNCs) Azarafrooz (2022) and Self-Attentive Associative Memory (SAM) Le et al. (2020) retain historical representations. Transformer-based approaches such as Transformer-XL (TrXL) Dai et al. (2019) and Scene Memory Transformer (SMT) Fang et al. (2019) further model dependencies across past states. These approaches primarily focus on storing, summarizing, or retrieving historical information. In contrast, we ask whether past experience can be used to *construct the representation of each new experience*. We propose *Contextual Episodic Memory (CEM)*, which progressively constructs episodic memory during navigation. Informative observations are selected online and encoded with the agent's 3D pose to form a visual-spatial representation h_t. Rather than storing h_t directly, CEM attends to previously stored representations to obtain contextual information c_t, which is combined with h_t to produce a contextualized representation $\tilde{h}_t$. This representation is then stored as the next episodic memory entry. Consequently, each new memory entry incorporates information from experiences accumulated earlier in the trajectory, allowing the memory to evolve as navigation progresses. The resulting memory provides the navigation policy with access to accumulated experience alongside the current observation. At each step, the policy can therefore associate the current view with information acquired earlier rather than

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

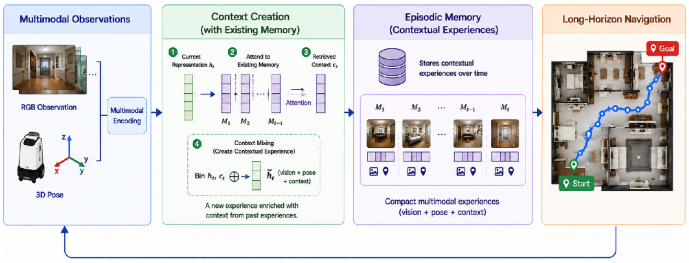


Figure 1: Overview of Contextual Episodic Memory (CEM). An informative visual-spatial observation attends to previously stored experiences to obtain contextual information, which is combined with the current representation and stored as the next episodic memory entry. This process progressively constructs an evolving representation of the navigation trajectory.

relying on isolated observations. CEM thus treats episodic memory not simply as a repository of past observations, but as an evolving representation of the navigation trajectory. We evaluate CEM in the photorealistic **AI2-THOR** environment Kolve et al. (2017) on two complementary navigation tasks. In *object search*, the agent must remember a sequence of rooms, observations, and spatial cues to locate a target object. In *room navigation*, it must retain a sequence of scene observations and transitions to reach a target room from different starting locations. Thus, both tasks require agents to accumulate and reuse information across a sequence of observations rather than act from isolated views. We compare CEM with recurrent, external-memory, and Transformer-based policies using navigation success, trajectory efficiency, memory usage, computational cost, and generalization to unseen environments. We further analyze the contributions of informative observation selection and contextual memory construction.

The contributions are as follows:

1. **Contextualized episodic memory.:** We introduce CEM, a read–contextualize–write memory mechanism in which previously accumulated experience is incorporated into the representation of the current experience before it is stored, enabling earlier experiences to causally influence subsequent episodic representations.
2. **Contextual Experience Construction:** CEM attends to previously stored representations, integrates the retrieved context with the current representation, and stores the resulting contextualized experience.
3. **Evaluation Across Navigation Tasks:** We evaluate CEM on multi-room object search and target-room navigation in AI2-THOR, demonstrating its ability to exploit accumulated experience for object-level and scene-level navigation.

Unlike conventional memory mechanisms that retrieve or associate previously stored experiences, CEM uses accumulated experience to contextualize the representation of the next experience before storing it. Through this read–contextualize–write process, prior experiences influence the representation of subsequent episodic traces, creating a causal chain of contextualized experiences over time. Thus, CEM focuses not only on what is retrieved, but on how prior experience shapes newly encoded experience.

2 Related Work

Memory and representation learning are central to navigation under partial observability. Existing approaches address this challenge by learning representations from observation histories, maintaining external memory, reusing episodic experiences, or constructing explicit spatial representations. We discuss these directions in terms of how information acquired during navigation is represented and preserved.

Sequential Representation Learning. A common approach summarizes previous observations into a single representation. LSTM Vennerød et al. (2021) and GRU Chung et al. (2014) policies maintain temporal context through recurrent hidden states, but compress history into a fixed-dimensional state that can lose specific visual-spatial experiences as the trajectory grows. Transformer-based approaches offer a more expressive alternative: Transformer-XL (TrXL) Dai et al. (2019) extends contextual representation learning across segments, while Scene Memory Transformer (SMT) Fang et al. (2019) learns representations from visual scene histories for em-

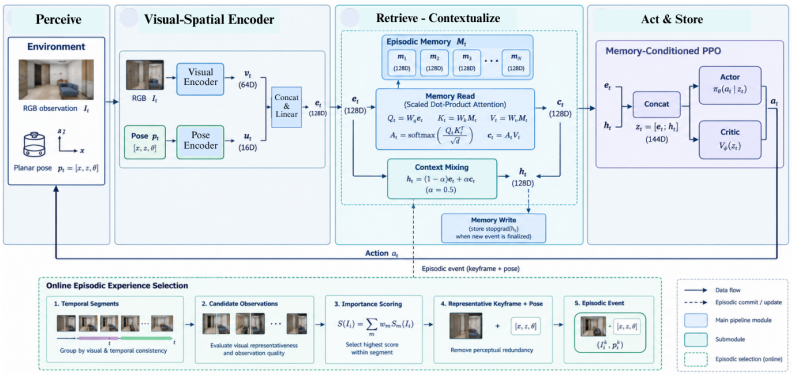


Figure 2: Overview of the proposed Contextual Episodic Memory (CEM) framework. RGB observations and planar poses are encoded into visual-spatial representations, with informative observations selected as keyframes. The current representation reads context from accumulated episodic memory, which is mixed with the current representation to form a context-enriched experience. This representation is used for policy inference and written back to memory, progressively building experience-based context for navigation.

bodied navigation. These methods primarily learn a representation of the observation sequence, using historical information as context for the current one.

External and Episodic Memory. Another line of work augments learned representations with explicit memory. Memory Networks Perez & Liu (2016); Burtsev et al. (2021), Differentiable Neural Computers (DNCs) Azarafrooz (2022), and Self-Attentive Associative Memory (SAM) **?** maintain external representations accessed through differentiable addressing or attention. Episodic memory has also been explored in reinforcement learning via Model-Free Episodic Control (MFEC) Blundell et al. (2016), Neural Episodic Control (NEC) Pritzel et al. (2017), and Continuous Episodic Control (CEC) Yang et al. (2023), which retain past experiences to support decision-making. These approaches show the value of preserving specific experiences beyond a compact recurrent state, but generally treat memory as stored information to be accessed or reused, rather than progressively using accumulated experience to construct the representation of a new one.

Memory for Visual Navigation. Memory is especially important for visual navigation since relevant information can disappear from the agent's current observation as it moves. Existing methods commonly maintain explicit spatial representations, including occupancy, semantic, and topological maps Gupta et al. (2017); Zhang et al. (2020); Gordon et al. (2018); Chaplot et al. (2018); Jiang et al. (2024); Jonschkowski et al. (2018); Savinov et al. (2018), which provide structured environmental information but typically require explicit mapping and localization. Others learn implicit memory through recurrent or attention-based representations Bonin-Font et al. (2008); Desouza & Kak (2002); Jefferies & Yeap (2008); Davison (2003); Dayoub et al. (2013); Sim & Little (2006); Zhao et al. (2023); Anderson et al. (2018); Kempka et al. (2016); Mirowski et al. (2017); Xia et al. (2018); Zhu et al. (2017; 2016), avoiding explicit maps but potentially compressing observations into representations that lose individual visual-spatial experiences.

CEM instead progressively constructs an episodic representation of the agent's experience: informative observations are selected online and encoded with the robot's 3D pose, each current representation attends to accumulated memory to retrieve contextual information, and the resulting contextualized representation is stored as the next memory entry, letting the memory evolve throughout navigation. Unlike methods that compress histories into recurrent states, directly retrieve past experiences, or build explicit spatial maps, CEM progressively constructs memory through contextual representation learning, with each stored entry integrating the current experience with information accumulated earlier.

3 METHODOLOGY

We propose an *episodic context-based memory framework* for long-horizon robotic object-search navigation under partial observability. The framework converts continuous observations into compact visual-spatial episodic experiences by selecting representative observations, associating them with robot poses, and encoding them into a shared representation space. During navigation, the current observation recalls relevant experiences, which are combined with the current state through

a context-mixing operation to produce a context-enriched representation for the PPO actor and critic. When an episodic event is finalized, this representation is committed to memory, allowing the agent to progressively accumulate context-enriched experiences rather than raw observations. An overview is shown in Fig. 2.

3.1 Online Episodic Experience Selection

Storing an experience at every navigation step would introduce substantial redundancy, as consecutive observations often contain similar visual information. We therefore construct episodic memory online by selecting representative observations that capture meaningful visual changes. For an RGB observation I_t, candidates within local temporal segments are scored as $S(I_i) = \sum_m w_m S_m(I_i)$ using multiple selection criteria. The highest-scoring observation is retained, followed by redundancy removal in the learned visual representation space. The resulting memory is $\mathcal{K}_t = \{(I_1^k, p_1^k), \ldots, (I_M^k, p_M^k)\}$, where each episodic event stores a selected keyframe I_i^k and its associated robot pose p_i^k. Thus, the memory retains informative visual-spatial experiences rather than an experience from every navigation step. Further details are provided in Appendix A.4.

3.2 Visual-Spatial Representation

For each decision step, the agent encodes its current RGB observation I_t and planar pose $p_t = [x_t, z_t, \theta_t]$ into compact latent representations, $v_t = f_{\text{vis}}(I_t)$ and $s_t = f_{\text{pose}}(p_t)$, respectively. These representations are fused into a shared visual-spatial embedding $e_t = W_f[v_t; s_t] + b_f$, where $e_t \in \mathbb{R}^{128}$ and $[\cdot;\cdot]$ denotes concatenation. The same multimodal encoder is applied to each selected keyframe and its associated pose, yielding $e_i^k = f_{\text{mm}}(I_i^k, p_i^k)$, where f_{mm} denotes the complete encoder. Consequently, current and episodic visual-spatial states are mapped into a common representation space, enabling direct comparison during memory retrieval.

3.3 Contextual Episodic Memory

CEM maintains an episodic memory of visual-spatial representations accumulated throughout navigation: $\mathcal{M}_t = \{m_1, m_2, \ldots, m_N\}$, where $m_i \in \mathbb{R}^{128}$.

Rather than storing each observation independently, CEM uses a *read–contextualize–write* operation to progressively construct the episodic memory. Given the current visual-spatial representation e_t, the existing memory $\mathcal{M}_t$ is first read to obtain contextual information from previously stored experiences. The current experience is then combined with this contextual information, and the resulting contextualized representation is written back to memory. Thus, each memory entry represents the current experience together with information accumulated from earlier experiences.

3.3.1 Memory Read

At time t, the current visual-spatial representation is used to read context from the accumulated episodic memory. We compute a query $Q_t = W_q e_t$, while the stored memory representations are projected into keys and values as $K_t = W_k \mathcal{M}_t$ and $V_t = W_v \mathcal{M}_t$.

The correspondence between the current experience and previously stored experiences is computed using scaled dot-product attention: $A_t = \text{softmax}\left(\frac{Q_t K_t^\top}{\sqrt{d}}\right)$,

where d is the attention dimension. The memory read produces a contextual representation $c_t = A_t V_t$.

The read operation therefore does not select a single previous observation. Instead, it forms a weighted contextual representation from the accumulated episodic experiences.

3.3.2 Context Creation and Memory Write

The memory-read context c_t is combined with the current representation e_t to create a contextualized representation:

$$h_t = (1 - \alpha)e_t + \alpha c_t, \tag{1}$$

where $\alpha \in [0, 1]$ controls the contribution of accumulated episodic context. In our implementation, $\alpha = 0.5$.

The resulting h_t is the representation that is written to episodic memory. When the current observation constitutes a new episodic event, we perform

$$\mathcal{M}_{t+1} = \mathcal{M}_t \cup \{\text{stopgrad}(h_t)\}. \tag{2}$$

Thus, the memory write does not store the raw representation e_t. Instead, it stores the representation obtained after combining the current experience with contextual information read from previously accumulated memory. This allows the memory to progressively incorporate information acquired throughout the navigation episode. The complete memory update can therefore be summarized as $\mathcal{M} * t \xrightarrow{\text{read}} c_t \xrightarrow{\text{context mixing with } e_t} h_t \xrightarrow{\text{write}} \mathcal{M} * t + 1$. This process is repeated as the agent encounters new informative experiences. Consequently, an experience stored at a later point in the trajectory is contextualized by representations accumulated earlier, while the newly stored representation becomes part of the context available for subsequent experiences.

3.4 Memory-Conditioned Navigation Policy

We form the policy representation as $z_t = [e_t; h_t]$, where e_t denotes the current visual-spatial observation and h_t its contextualized representation with the accumulated episodic memory. The actor and critic operate on z_t: $\pi_\theta(a_t|z_t) = \text{Categorical}(\text{logits}(f_{\text{actor}}(z_t)))$ and $V_\phi(z_t) = f_{\text{critic}}(z_t)$. This enables navigation decisions to use both the current state and episodic context.

4 Experimental Setup

Experimental Scenarios

Multi-Room Object Search. The agent navigates through multiple rooms to discover five target objects: `Television`, `HousePlant`, `Sofa`, `CoffeeMachine`, and `Fridge`. Objects may be encountered before they become visible again during subsequent navigation. This scenario therefore evaluates whether the episodic memory preserves contextualized visual-spatial experiences acquired earlier in the episode and enables the agent to use them when making later navigation decisions.

Target-Room Navigation. The agent is provided with a visual target corresponding to a kitchen and must navigate toward the target room from randomly sampled reachable locations. The target is not continuously visible during navigation, requiring the agent to use previously accumulated visual-spatial context when selecting subsequent actions. We evaluate the trained policy in a separate AI2-THOR scene with different room layouts, object configurations, visual appearance, and lighting, without retraining or scene-specific adaptation.

Implementation Details We evaluate CEM in AI2-THOR on the `ArchitecTHOR-Val-01` scene set. RGB observations are rendered at 800×600 and resized to 224×224. The agent operates on a 0.5-m navigation grid with ten discrete actions and receives only RGB observations and planar pose (x, z, θ); depth and semantic maps are not provided to the policy. An object is considered found within 1.5 m.

RGB observations are encoded using ImageNet-pretrained ResNet-18 with a 64-D output, while pose is encoded by a $3 \to 32 \to 16$ MLP. These features are fused into a 128-D visual-spatial representation e_i. The current representation reads from episodic memory to obtain c_i, which is mixed with e_i to produce h_i and subsequently written back to memory:

$$\mathcal{M} * i \xrightarrow{\text{read}} c_i \xrightarrow{\text{mix}} h_i \xrightarrow{\text{write}} \mathcal{M} * i + 1. \tag{3}$$

The policy input is $z_i = [e_i; h_i]$, processed by separate actor and critic MLPs and optimized using PPO. We report Success Rate, cumulative reward, episode length, and trajectory path length. Further implementation and evaluation details are provided in Appendix A.

5 Experiments and Results

5.1 Contextual Memory Generalization for Object Search

We evaluate whether CEM generalizes object-search navigation as the correspondence between training and test environments is progressively disrupted, motivated by settings where the current observation alone is insufficient to locate a target object. We construct three test conditions that preserve the underlying search task while introducing increasing visual and contextual variation.

Evaluation Protocol. The agent is trained in a reference environment and evaluated in modified versions of it. For each episode, the initial spawn position is randomized and a target object is specified as the search goal. Test environments progressively weaken direct visual correspondence with training while preserving useful relationships among objects, locations, and context.

Table 1: Object-search accuracy (%) in the reference training environment and three modified test environments, reported as mean ± standard deviation over 5 seeds. Higher is better.

Model	Train	Test 1	Test 2	Test 3
CEM (Ours)	**86.0 ± 2.1**	**82.0 ± 2.6**	**60.0 ± 3.4**	**64.0 ± 3.1**
DNC	83.0 ± 3.0	65.0 ± 3.8	49.0 ± 4.2	51.0 ± 4.0
SAM	85.0 ± 2.4	79.0 ± 3.0	55.0 ± 3.6	58.0 ± 3.3
TrXL	63.0 ± 4.5	42.0 ± 5.1	45.0 ± 4.8	48.0 ± 4.6
LSTM	56.0 ± 5.0	31.0 ± 5.5	35.0 ± 5.2	38.0 ± 5.0
No Memory	45.0 ± 5.8	23.0 ± 6.0	21.0 ± 5.9	26.0 ± 5.7

Table 2: Average navigation path length across Test 1–3. Values are mean ± standard deviation in meters; lower is better.

Method	Test 1 (m)	Test 2 (m)	Test 3 (m)
CEM (Ours)	71.03 ± 6.27	74.25 ± 6.58	77.31 ± 6.98
TrXL	93.33 ± 8.30	96.84 ± 8.80	100.85 ± 9.10
DNC	100.33 ± 9.30	103.73 ± 9.58	107.85 ± 9.99
SAM	104.46 ± 9.79	108.24 ± 10.14	112.40 ± 10.51
LSTM	117.04 ± 11.10	121.33 ± 11.47	126.57 ± 11.92
No Memory	95.60 ± 13.10	100.43 ± 21.66	134.57 ± 21.92

Contextual Perturbations. **Test 1** changes scene colors and relocates selected objects, disrupting appearance–location correspondences while preserving overall scene structure, testing robustness to moderate visual and spatial change. **Test 2** additionally alters selected room semantics, requiring experience to generalize beyond direct visual similarity. **Test 3** further introduces duplicate object instances, so visually similar candidates occur at different locations and the agent must use contextual information rather than appearance alone to disambiguate.

Together, these conditions provide a controlled progression from visual variation to contextual ambiguity, testing whether episodic experiences support search when previously observed configurations no longer match the current environment. Detailed environment configurations are in Appendix A.1.

Table 1 reports object-search accuracy (mean ± std over 5 seeds) against DNC, SAM, TrXL, LSTM, and a no-memory baseline.

CEM achieves the highest accuracy in the reference environment and across all three modified environments, decreasing as environmental correspondence weakens but remaining substantially above the no-memory baseline and competing architectures. Notably, CEM retains $64.0\% \pm 3.1$ accuracy in Test 3, where duplicate objects introduce ambiguity unresolvable from appearance alone. The margin over SAM is narrow in the reference environment (86.0% vs. 85.0%) but widens under stronger perturbations (Test 2: 60.0% vs. 55.0%; Test 3: 64.0% vs. 58.0%), with CEM achieves higher mean accuracy than SAM across all evaluated settings, with the largest differences occurring under the stronger environmental perturbations. These results indicate that CEM's contextualized experiences provide information beyond the current visual observation, supporting search when previously encountered configurations have changed.

5.1.1 Spawn Position and Trajectory Analysis

We further examine generalization across initial spawn positions by measuring navigation path length in the three modified environments. Varying spawn position changes the observation sequence encountered during navigation, complementing the accuracy results by testing whether stored experiences remain useful along unfamiliar trajectories; shorter paths indicate less exploratory navigation. CEM produces the shortest trajectories in all three settings (71.03, 74.25, and 77.31 m for Test 1–3), with path length increasing alongside environmental variation while small standard deviations indicate stable navigation across randomized spawn positions. Together with the accuracy results, these measurements capture both successful target discovery and the exploration required under changed environmental contexts.

5.2 Object Removal and Contextual Episodic Recall

Real-world environments are rarely static: objects may be moved, occluded, or removed before the agent must act on prior knowledge. Episodic memory enables the agent to retain *where* an object

Table 3: Episodic recall under object removal. Recall denotes the percentage of evaluation episodes in which the agent successfully revisits the original location of a target object after the object is removed from the environment. Higher is better.

Method	**Recall (%)**
LSTM	40.25
Recurrent Transformer	55.42
Transformer-XL (TrXL)	46.89
Differential Neural Computer (DNC)	56.75
Self-attentive Associative Memory (SAM)	60.24
CEM (Ours)	**65.89**

Table 4: Target-room navigation accuracy (%) on unseen environments. Higher is better. Best results are shown in bold.

Method	**Env. 1**	**Env. 2**	**Env. 3**	**Env. 4**	**Average**
Ours	**86.92**	**77.18**	**79.35**	**72.31**	**78.94**
SAM	85.29	74.62	75.00	71.25	76.54
DNC	80.00	60.38	58.00	68.92	66.83
TrXL	65.00	53.74	54.00	62.34	58.77
LSTM	40.00	45.21	44.00	45.24	43.61

was previously observed and revisit its location even when it is no longer visible. We evaluate this capability through an object-removal experiment in which target objects are removed at test time while the room layout, geometry, and obstacles remain unchanged. The agent must therefore navigate to the object's original location using only its memory. We measure *episodic recall* as the percentage of episodes in which the agent successfully revisits this location. Table 3 compares CEM with recurrent, Transformer-based, and external-memory baselines. CEM achieves the highest recall at 65.89%, outperforming all baselines. Since the task removes the current visual cue, successful recall requires memory to encode location-specific spatial context rather than simply whether an object was previously observed. CEM's advantage over external-memory baselines suggests that contextualizing retrieved experiences, rather than merely storing and retrieving past observations, is important for preserving this information. Overall, the results highlight spatial memory as a key component of navigation under partial observability.

5.3 Generalization to Unseen Environments

To evaluate transfer beyond the training distribution, we test the agent in four previously unseen environments with varying appearance, illumination, and spatial layout, including darker regions and different room configurations. Given an RGB observation of the target room, the agent must navigate there from a randomized start location, requiring it to exploit target-relevant visual cues while adapting to conditions absent from training. This setting is informative because a room's appearance can vary substantially across environments while its semantic identity stays fixed, testing whether the agent relies on environment-specific memorization or transferable visual experience. Tables 4 and 5 summarize accuracy and trajectory length across the four unseen environments. CEM maintains consistent performance under illumination, appearance, and layout changes, indicating that its episodic context provides transferable visual-spatial cues for navigation. Step-efficiency results are provided in Appendix B.4.

Memory-to-target similarity analysis To examine how different memory architectures preserve target-relevant information over an episode, Fig. 3 visualizes the cosine similarity between stored memory representations and the target representation across time. CEM exhibits distinct high-similarity regions at multiple points in the trajectory, indicating that target-relevant information remains represented in episodic experiences beyond only the most recent observations. LSTM and Transformer-XL show predominantly localized similarity near the current timestep, whereas DNC and SAM exhibit broader and more diffuse target-related similarity patterns. This reflects a different memory organization pattern: CEM progressively incorporates previously accumulated context into newly stored episodic representations, preserving target-relevant information within selected experiences rather than making the entire memory broadly target-aligned. Such selective organization is desirable for episodic memory, whose purpose is to preserve context-rich representations of in-

Table 5: Navigation path length (m) across unseen environments, reported as mean ± standard deviation. Lower values indicate shorter paths. The best result in each environment is shown in bold.

Method	Env. 1	Env. 2	Env. 3	Env. 4	Average
Ours	**13.28 ± 11.64**	**17.36 ± 25.81**	**25.41 ± 25.37**	**29.63 ± 29.14**	**21.42**
SAM	19.26 ± 24.18	23.85 ± 12.67	28.72 ± 47.35	33.26 ± 51.42	26.27
DNC	14.83 ± 11.46	31.46 ± 28.35	61.45 ± 58.24	69.35 ± 63.28	44.27
TrXL	47.82 ± 34.25	56.35 ± 41.28	63.82 ± 69.35	74.28 ± 75.46	60.57
LSTM	62.35 ± 46.28	185.62 ± 28.46	91.28 ± 48.62	108.62 ± 55.37	112.97

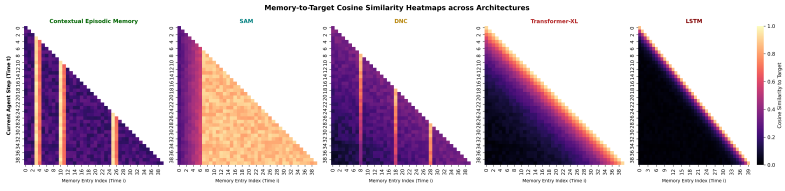


Figure 3: Memory-to-target cosine similarity across time for different memory architectures. Each heatmap shows the cosine similarity between stored memory representations and the target representation at each agent time step. CEM exhibits temporally distributed, selective high-similarity regions, indicating retrieval of target-relevant episodic experiences beyond the most recent observation, while recurrent and transformer-based baselines show more localized or diffuse similarity patterns.

dividual experiences while retaining target-relevant information that can be accessed when needed, rather than transforming the entire memory into a target-specific representation. Thus, Fig. 3 provides a qualitative view of how CEM organizes episodic experiences over time, complementary to the quantitative navigation and recall results.

Qualitative Analysis of Memory and Navigation. Figure 4 provides a qualitative comparison between memory activation patterns and the resulting navigation trajectories. The proposed method activates a small number of episodic memory units at relevant time steps and produces a relatively direct trajectory toward the target. In contrast, the baseline methods show more distributed or repeated memory activation and correspondingly longer or more redundant trajectories. This qualitative behavior is consistent with the quantitative results in Tables 4 and 5.

5.4 Keyframe Contribution to Episodic Navigation

We analyze the spatial distribution, target relevance, and navigation impact of selected keyframes. Figure 5(a) shows a representative refrigerator-search episode, where cyan markers $K_1, \ldots, K_{10}$ denote selected keyframes along the navigation trajectory. Each keyframe stores a visual observation with its corresponding pose. Figure 5(b) shows that early keyframes capture general scene context, while K_4–K_6 exhibit stronger target similarity and retrieval attention. Later keyframes provide additional spatial context for re-localization. As shown in Fig. 5(c), contextualized retrieval produces the largest reduction in target distance around K_4–K_7, indicating that target-relevant episodic experiences provide actionable navigation cues.

Table 6: Ablation on the memory representation used for contextualization. All configurations use the same contextual mixing mechanism; only the memory granularity is varied. Values denote success rates (%).

Memory Representation	Seen	Unseen	Removal	Avg.
All observations	78	72	58	69.3
Level-1 keyframe	74	66	52	64.0
Level-2	**86**	**82**	**64**	**77.3**

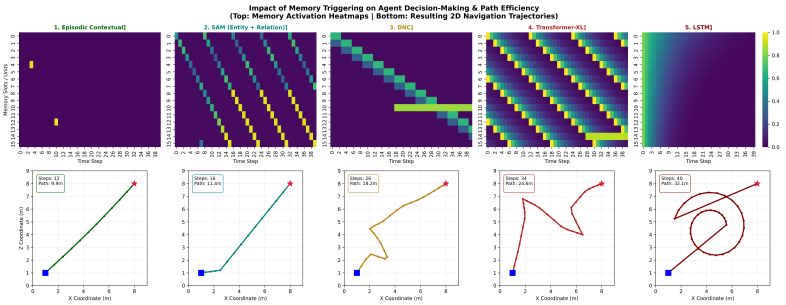
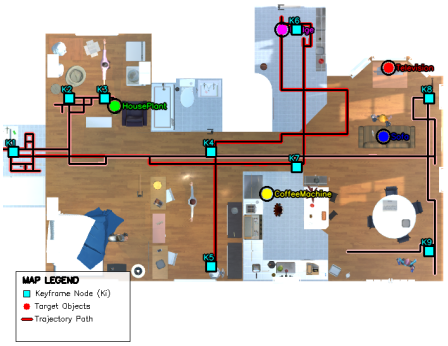
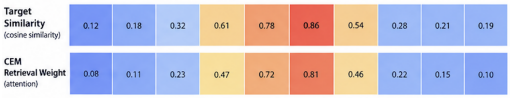


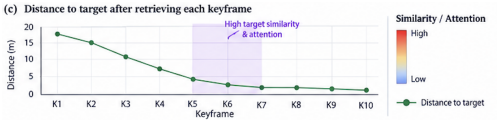
Figure 4: Qualitative comparison of memory activation and navigation behavior across different memory architectures. Top: memory activation patterns over time. Bottom: corresponding 2D navigation trajectories, with the blue square denoting the starting position and the red star denoting the target. The proposed episodic contextual memory produces sparse memory activation and a direct trajectory, whereas recurrent and transformer-based baselines exhibit more distributed activation and longer, more redundant trajectories.



(a) Trajectory and selected keyframes K_1–K_{10}.



(b) Target similarity and CEM retrieval weight per keyframe.



(c) Distance to target after retrieving each keyframe.

Figure 5: Visualization of CEM's keyframe retrieval and navigation behavior.

Ablation on memory granularity. We vary the memory representation while keeping the contextual mixing mechanism fixed. Using all observations, Level-1 keyframes, and Level-2 keyframes yields average success rates of 69.3%, 64.0%, and 70.0%, respectively. The full Level-2+CEM configuration achieves 77.3%, indicating that selective episodic representations provide useful context while avoiding redundant observations. Results are reported in Table 6.

Effect of contextual mixing. We vary the mixing coefficient α, which controls the contribution of episodic context. Performance increases from 68.1% at $\alpha = 0$ to 77.3% at $\alpha = 0.5$, then decreases to 70.2% at $\alpha = 1.0$. This indicates that combining current perceptual evidence with episodic context is more effective than relying exclusively on either source.

6 CONCLUSION

We presented Contextual Episodic Memory (CEM), a framework that progressively constructs episodic memory by integrating retrieved context into newly encountered observations before writing them back to memory. This produces context-enriched experiences that support navigation beyond isolated past observations. Across multi-room object search and target-room navigation in AI2-THOR, CEM consistently outperformed recurrent, external-memory, and Transformer-based baselines across seen, unseen, and object-removal settings. It achieved higher navigation accuracy, more effective recall of previously observed targets, and shorter trajectories under changes in layout, appearance, and lighting. Analysis further shows that CEM retrieves temporally distributed, target-relevant experiences rather than relying only on recent observations. Ablations indicate that contextual read–contextualize–write contributes more substantially than memory granularity alone, improving average success from 70.0% with Level-2 keyframes to 77.3% with CEM. Overall, the results highlight that effective episodic navigation depends not only on which experiences are retained, but also on how they are progressively contextualized.

AI USE STATEMENT

We used AI-assisted language tools to improve the clarity, grammar, and overall presentation of the manuscript, including rephrasing and proofreading of the written content. AI tools were also used to assist with the conceptual design and refinement of methodology diagrams. All scientific content, methodological decisions, experimental design, results, and conclusions were developed and verified by the authors, who remain fully responsible for the final manuscript.

REPRODUCIBILITY STATEMENT

Reproducibility Statement. We provide detailed descriptions of the experimental setup, datasets, implementation details, model configurations, training procedures, and evaluation protocols in the main paper and Appendix to facilitate reproducibility of our results. Most of the experiments can be reproduced using the information and procedures provided in the Appendix. To further support reproducibility, we plan to release the source code and relevant implementation details publicly upon acceptance of the paper.

REFERENCES

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments, 2018. URL https://arxiv.org/abs/1711.07280.

Ari Azarafrooz. Differentiable neural computers with memory demon, 2022. URL https://arxiv.org/abs/2211.02987.

Charles Blundell, Benigno Uria, Alexander Pritzel, Yazhe Li, Avraham Ruderman, Joel Z Leibo, Jack Rae, Daan Wierstra, and Demis Hassabis. Model-free episodic control, 2016. URL https://arxiv.org/abs/1606.04460.

Francisco Bonin-Font, Alberto Ortiz, and Gabriel Oliver. Visual navigation for mobile robots: A survey. *Journal of Intelligent and Robotic Systems*, 53:263–296, 11 2008. doi: 10.1007/s10846-008-9235-4.

Mikhail S. Burtsev, Yuri Kuratov, Anton Peganov, and Grigory V. Sapunov. Memory transformer, 2021. URL https://arxiv.org/abs/2006.11527.

Devendra Singh Chaplot, Emilio Parisotto, and Ruslan Salakhutdinov. Active neural localization, 2018. URL https://arxiv.org/abs/1801.08214.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014. URL https://arxiv.org/abs/1412.3555.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context, 2019. URL https://arxiv.org/abs/1901.02860.

Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pp. 1403–1410 vol.2, 2003. doi: 10.1109/ICCV.2003.1238654.

Feras Dayoub, Timothy Morris, Ben Upcroft, and Peter Corke. Vision-only autonomous navigation using topometric maps. pp. 1923–1929, 11 2013. doi: 10.1109/IROS.2013.6696611.

G.N. Desouza and A.C. Kak. Vision for mobile robot navigation: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267, 2002. doi: 10.1109/34.982903.

Kuan Fang, Alexander Toshev, Li Fei-Fei, and Silvio Savarese. Scene memory transformer for embodied agents in long-horizon tasks, 2019. URL https://arxiv.org/abs/1903.03878.

Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. Iqa: Visual question answering in interactive environments, 2018. URL https://arxiv.org/abs/1712.03316.

Saurabh Gupta, David Fouhey, Sergey Levine, and Jitendra Malik. Unifying map and landmark based representations for visual navigation, 2017. URL https://arxiv.org/abs/1712.08125.

Margaret Jefferies and Wai-Kiang Yeap. *Robotics and Cognitive Approaches to Spatial Mapping*, volume 38. 01 2008. ISBN 978-3-540-75386-5. doi: 10.1007/978-3-540-75388-9.

Zhou Jiang, Zhenxin Zhu, Pengfei Li, Huan ang Gao, Tianyuan Yuan, Yongliang Shi, Hang Zhao, and Hao Zhao. P-mapnet: Far-seeing map generator enhanced by both sdmap and hdmap priors, 2024. URL https://arxiv.org/abs/2403.10521.

Rico Jonschkowski, Divyam Rastogi, and Oliver Brock. Differentiable particle filters: End-to-end learning with algorithmic priors, 2018. URL https://arxiv.org/abs/1805.11122.

Michał Kempka, Marek Wydmuch, Grzegorz Runc, Jakub Toczek, and Wojciech Jaśkowski. Vizdoom: A doom-based ai research platform for visual reinforcement learning, 2016. URL https://arxiv.org/abs/1605.02097.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017.

Hung Le, Truyen Tran, and Svetha Venkatesh. Self-attentive associative memory, 2020. URL https://arxiv.org/abs/2002.03519.

Piotr Mirowski, Razvan Pascanu, and Fabio Viola. Learning to navigate in complex environments, 2017. URL https://arxiv.org/abs/1611.03673.

Julien Perez and Fei Liu. Gated end-to-end memory networks, 2016. URL https://arxiv.org/abs/1610.04211.

Alexander Pritzel, Benigno Uria, Sriram Srinivasan, Adrià Puigdomènech, Oriol Vinyals, Demis Hassabis, Daan Wierstra, and Charles Blundell. Neural episodic control, 2017. URL https://arxiv.org/abs/1703.01988.

Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation, 2018. URL https://arxiv.org/abs/1803.00653.

Robert Sim and J.J. Little. Autonomous vision-based exploration and mapping using hybrid maps and rao-blackwellised particle filters. pp. 2082 – 2089, 11 2006. doi: 10.1109/IROS.2006.282485.

Christian Bakke Vennerød, Adrian Kjærran, and Erling Stray Bugge. Long short-term memory rnn, 2021. URL https://arxiv.org/abs/2105.06756.

Fei Xia, Amir Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents, 2018. URL https://arxiv.org/abs/1808.10654.

Zhao Yang, Thomas M. Moerland, Mike Preuss, and Aske Plaat. Continuous episodic control, 2023. URL https://arxiv.org/abs/2211.15183.

Jingwei Zhang, Lei Tai, Ming Liu, Joschka Boedecker, and Wolfram Burgard. Neural slam: Learning to explore with external memory, 2020. URL https://arxiv.org/abs/1706.09520.

Qianfan Zhao, Lu Zhang, Bin He, Hong Qiao, and Zhiyong Liu. Zero-shot object goal visual navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2025–2031, 2023. doi: 10.1109/ICRA48891.2023.10161289.

Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, and Joseph J. Lim. Target-driven visual navigation in indoor scenes using deep reinforcement learning, 2016. URL https://arxiv.org/abs/1609.05143.

Yuke Zhu, Daniel Gordon, Eric Kolve, Dieter Fox, Li Fei-Fei, Abhinav Gupta, Roozbeh Mottaghi, and Ali Farhadi. Visual semantic planning using deep successor representations, 2017. URL https://arxiv.org/abs/1705.08080.

.1 LEVEL-1 AND LEVEL-2 KEYFRAME CONSTRUCTION

This section details the two-level keyframe hierarchy used in the memory-granularity ablation (Section 5.4). Level-1 keyframes are obtained directly from causal boundary detection and temporal consolidation, while Level-2 keyframes apply a further retrieval-aware selection step on top of the Level-1 output to produce a more compact and discriminative set of anchors. Both levels are constructed online, using only past and current observations, consistent with the causal setting of episodic memory construction described in Section 5.4.

Stage 1: Perceptual boundary detection (Level-1, coarse). Given the incoming observation stream $I_1, I_2, \ldots, I_T$, each frame I_t is mapped to a perceptual embedding z_t using a lightweight visual encoder. A temporal boundary is declared whenever the embedding distance between consecutive frames exceeds a fixed threshold τ_c:

$$b_t = \begin{cases} 1, & \|z_{t+1} - z_t\|_2 > \tau_c, \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

This partitions the stream into an ordered sequence of temporal segments $S = \{s_1, \ldots, s_K\}$. Within each segment $s_k = [a_k, b_k]$, a coarse representative frame is chosen as the one closest to the segment's reference (first) frame in embedding space:

$$d_i = \|z_i - z_{a_k}\|_2, \qquad \hat{k}_k = \arg\min_{i \in [a_k, b_k]} d_i. \tag{5}$$

Consecutive segments separated by a small temporal gap $g_i = a_{i+1} - b_i \leq \gamma$ are merged to reduce over-segmentation caused by transient motion blur or brief viewpoint changes. The resulting coarse anchors, one per consolidated segment, constitute the **Level-1 keyframe set** $\mathcal{K}^{(1)} = \{\hat{k}_1, \ldots, \hat{k}_K\}$. Level-1 keyframes therefore capture perceptually coherent temporal boundaries but are selected using only local segment statistics, without explicit regard to redundancy across the full memory or downstream retrieval utility.

Stage 2: Retrieval-aware optimization (Level-2, refined). Level-2 keyframes are obtained by re-ranking candidates within each consolidated segment using a retrieval-aware utility that jointly considers semantic centrality, visual informativeness, and distinctiveness from previously stored memory.

For each candidate frame f_i in a consolidated group G_m, we compute a semantic embedding $x_i = \phi(f_i)$ and the group centroid

$$\mu_m = \frac{1}{|G_m|} \sum_{f_i \in G_m} x_i. \tag{6}$$

A candidate pool C_m is formed from the frames closest to μ_m (the nearest $p\%$), discarding outlier observations while retaining representative candidates. The utility of each candidate is

$$U(f_i) = \alpha_1 \, C(f_i) + \beta \, I(f_i) + \lambda \, D(f_i), \tag{7}$$

where semantic centrality favors frames near the group centroid,

$$C(f_i) = -\|x_i - \mu_m\|_2, \tag{8}$$

visual informativeness favors frames with richer spatial structure,

$$I(f_i) = \mathrm{Var}(\nabla^2 f_i) \cdot H(f_i), \tag{9}$$

with $\mathrm{Var}(\nabla^2 f_i)$ the Laplacian (high-frequency) response and $H(f_i)$ the grayscale entropy, and distinctiveness penalizes redundancy with already-stored memory entries,

$$D(f_i) = \min_{k_j \in \mathcal{K}^{(2)}_{<m}} \|x_i - x_{k_j}\|_2. \tag{10}$$

The Level-2 keyframe for group G_m is the maximizer of this utility,

$$k_m = \arg\max_{f_i \in C_m} U(f_i), \tag{11}$$

yielding the **Level-2 keyframe set** $\mathcal{K}^{(2)} = \{k_1, \ldots, k_M\}$, $M \leq K$. Because $D(\cdot)$ is evaluated against previously selected Level-2 anchors, this stage explicitly enforces inter-memory diversity across the whole episode rather than only within a single segment, which is the key distinction from Level-1 selection.

Implementation details. Boundary detection (Stage 1) uses content-based scene segmentation with a threshold of $\tau_c = 27$, applied to frames sampled at approximately 2 fps. Segment-level distances are computed using DreamSim embeddings, and adjacent segments separated by fewer than 3 frames are merged ($\gamma = 3$). For Level-2 refinement, candidate frames within each consolidated group are encoded with a pretrained ResNet-18 (matching the visual encoder used elsewhere in the pipeline; a ResNet-50 backbone was also evaluated with similar trends), and the nearest 50% of frames to the group centroid form the candidate pool C_m. Visual informativeness is computed from Laplacian variance and grayscale entropy, and the utility weights are set empirically to $\alpha_1 = \beta = \lambda = 1/3$ after a small grid search. Frames with Laplacian variance below 80 are discarded prior to scoring, consistent with the quality filtering described in Appendix A.4.

Relation to memory granularity ablation. Level-1 keyframes provide a purely boundary-driven summary of the trajectory and act as a lower bound on selection quality, since no cross-segment redundancy or retrieval utility is considered. Level-2 keyframes apply the additional retrieval-aware optimization above, producing a more compact and mutually distinctive memory. As shown in Table 6, this refinement alone improves average success from 64.0% (Level-1) to 70.0% (Level-2), and the full CEM contextual retrieval mechanism applied on top of Level-2 keyframes further improves performance to 77.3%, indicating that boundary-aware selection and contextual memory construction are complementary rather than redundant sources of improvement.

A Implementation and Experimental Details

A.1 Environment and Navigation

All simulated experiments are conducted in AI2-THOR using the `ArchitecTHOR-Val-01` scene set. RGB observations are rendered at 800×600 pixels and resized to 224×224 before being processed by the visual encoder. The agent is initialized at a randomly sampled reachable position with heading 0° on a 0.5-m navigation grid.

The action space contains ten discrete actions: `MoveAhead`, `MoveBack`, `RotateLeft`, `RotateRight`, `LookUp`, `LookDown`, `OpenObject`, `CloseObject`, `MoveLeft`, and `MoveRight`. The navigation policy receives only the RGB observation and the planar pose (x, z, θ); depth and semantic maps are not provided as policy inputs. An object is considered found when the agent is within 1.5 m of the target.

A.2 Navigation Scenarios

A.2.1 Scenario 1: Multi-Room Object Search

In the object-search scenario, the agent must discover five target objects: `Television`, `HousePlant`, `Sofa`, `CoffeeMachine`, and `Fridge`. The agent navigates through multiple rooms while incrementally constructing its episodic memory.

This scenario is designed to evaluate whether the memory preserves visual-spatial information acquired earlier in the episode. In particular, an object or its surrounding scene may be observed at

one point in the trajectory and may no longer be visible from the current observation. The current observation can nevertheless be contextualized using the experiences accumulated in episodic memory. Thus, successful navigation requires the agent to use information accumulated over the course of the episode rather than relying only on the currently visible target.

An episode terminates when all five target objects have been discovered or when the maximum episode length of 2000 policy steps is reached.

A.2.2 SCENARIO 2: TARGET-ROOM NAVIGATION

In the target-room scenario, the agent is given a visual target corresponding to a kitchen and must navigate toward the target room. Starting locations are randomly sampled from reachable positions, and the agent approaches the target from different sides.

The target room is not continuously visible during navigation. The scenario therefore evaluates whether the episodic memory can preserve and contextualize visual-spatial experience accumulated before the target becomes visible, allowing the navigation policy to use previously encountered scene information when selecting subsequent actions.

For cross-scene evaluation, the policy is trained in the training environment and transferred to a separate AI2-THOR scene with different room layouts, object configurations, visual appearance, and lighting conditions. The target is specified by an RGB kitchen image from the new scene. No retraining or scene-specific adaptation is performed.

A.3 MULTIMODAL STATE ENCODER

The current RGB observation is encoded using an ImageNet-pretrained ResNet-18 with a 64-dimensional output. The planar pose $p_i = [x_i, z_i, \theta_i]$ is encoded using an MLP with dimensions $3 \to 32 \to 16$. The visual and pose features are concatenated and projected to a 128-dimensional visual-spatial representation:

$$e_i = W_f[v_i; u_i] + b_f, \qquad e_i \in \mathbb{R}^{128}. \tag{12}$$

The same encoder is used for observations selected for episodic memory. RGB values are scaled by $1/255$.

A.4 ONLINE EPISODIC EXPERIENCE SELECTION

Informative observations are selected online at temporal boundaries detected using grayscale pixel changes with a threshold of 27.0 or when the maximum segment length reaches 20 frames. Candidate frames are compared with the latest selected observation using DreamSim.

Selected observations are subsequently grouped by place using BLIP captions and spaCy entity extraction. Each finalized group is represented by the frame that maximizes a weighted image-quality score based on Laplacian variance, entropy, ResNet feature strength, brightness, and image area. The corresponding weights are $[0.35, 0.25, 0.20, 0.10, 0.10]$. Frames with Laplacian variance below 80 are discarded.

The resulting observation and its corresponding pose are encoded using the multimodal state encoder and form an episodic visual-spatial experience.

A.5 EPISODIC MEMORY READ AND WRITE

Let the episodic memory available when processing the current observation i be

$$\mathcal{M}_i = \{m_1, \ldots, m_N\}, \qquad m_j \in \mathbb{R}^{128}. \tag{13}$$

The current representation e_i is used to read contextual information from the accumulated memory:

$$Q_i = W_q e_i, \qquad K = W_k \mathcal{M}_i, \qquad V = W_v \mathcal{M}_i, \tag{14}$$

followed by scaled dot-product attention:

$$A_i = \mathrm{softmax}\left(\frac{Q_i K^\top}{\sqrt{d}}\right), \tag{15}$$

and

$$c_i = A_i V. \tag{16}$$

The read context is combined with the current representation to construct the contextualized experience:

$$h_i = (1 - \alpha)e_i + \alpha c_i, \tag{17}$$

where $\alpha = 0.5$ is fixed.

When the current observation corresponds to a newly finalized episodic event, the contextualized representation is written to memory:

$$\mathcal{M}_{i+1} = \mathcal{M}_i \cup \{\mathrm{stopgrad}(h_i)\}. \tag{18}$$

Thus, the memory operation can be summarized as

$$\boxed{\mathcal{M}_i \xrightarrow{\text{read}} c_i \xrightarrow{\text{context mixing with } e_i} h_i \xrightarrow{\text{write}} \mathcal{M}_{i+1}}. \tag{19}$$

Importantly, the representation written to memory is the contextualized representation h_i, rather than the independently encoded representation e_i. Consequently, newly stored experiences incorporate information from experiences accumulated earlier in the episode and subsequently become available for contextualizing future observations.

A.6 Memory-Conditioned Navigation Policy

The navigation policy receives both the current visual-spatial representation and its contextualized representation:

$$z_i = [e_i; h_i]. \tag{20}$$

The actor and critic use separate MLPs with dimensions $256 \to 256$ and ReLU activations. The actor predicts logits over the ten navigation actions, while the critic estimates the state value.

The actor is defined as

$$\pi_\theta(a_i|z_i) = \mathrm{Categorical}\left(\mathrm{logits}(f_{\mathrm{actor}}(z_i))\right), \tag{21}$$

and the critic is

$$V_\phi(z_i) = f_{\mathrm{critic}}(z_i). \tag{22}$$

When the episodic memory is empty, the memory-read operation is bypassed and the current representation is used without an accumulated memory context.

A.7 PPO Training

The memory-conditioned actor–critic is optimized using PPO. The learning rate is 10^{-4}, with discount factor $\gamma = 0.99$, GAE parameter $\lambda = 0.95$, and PPO clipping ratio 0.2. We use four optimization epochs per rollout and a batch size of 64. The value and entropy coefficients are 0.5 and 0.01, respectively, and gradients are clipped to a maximum norm of 0.5.

Rollouts contain 256 policy steps, and training uses a budget of 5×10^9 environment steps. Memory entries are detached before storage so that computation graphs are not propagated across subsequent memory updates.

A.8 Reward Function

The reward is composed of progress, milestone, penalty, loop, and success terms:

$$r_i = r_i^{\mathrm{prog}} + r_i^{\mathrm{mil}} + r_i^{\mathrm{pen}} + r_i^{\mathrm{loop}} + r_i^{\mathrm{succ}}. \tag{23}$$

The progress reward is

$$r_i^{\mathrm{prog}} = \begin{cases} 2.5(s_i - s_{i-1}), & s_i > s_{i-1},\ s_i > 0.4, \\ 0, & \text{otherwise}, \end{cases} \tag{24}$$

where

$$s_i = \cos(\mathbf{v}_i, \mathbf{g})$$

denotes the cosine similarity between the current and target embeddings.

A reward of 2.0 is assigned whenever a new one-decimal similarity milestone is crossed, up to a similarity of 0.8. The penalty term contains a -0.05 cost per simulator action, a -0.1 penalty for failed actions, a rotation penalty proportional to consecutive rotations, and a -0.5 penalty for immediately reversing a translational action.

To discourage visual loops, a penalty of -2.0 is applied when the current observation has cosine similarity greater than 0.98 with a recent observation, and a penalty of -1.0 is applied when the similarity lies in $(0.95, 0.98]$. Successful target reaching terminates the episode and provides a terminal reward of $+10$.

Target success is determined by target-object/place matching when available, or by target-image similarity exceeding 0.88 otherwise.

A.9 Evaluation Metrics

We report Success Rate (SR), cumulative episode reward, episode length, and trajectory path length. Success Rate is computed as

$$\mathrm{SR} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}[\mathrm{success}_i] \times 100. \tag{25}$$

Episode length denotes the number of policy steps until termination. Trajectory path length is computed from the recorded 3-D agent positions:

$$PL_i = \sum_{j=1}^{T_i} \|\mathbf{p}_{i,j} - \mathbf{p}_{i,j-1}\|_2 . \tag{26}$$

We report the mean and standard deviation of episode length and path length. The maximum episode length is 2000 policy steps.

A.10 Cross-Scene Generalization

For cross-scene evaluation, the trained policy is transferred to a separate AI2-THOR scene containing different room layouts, object configurations, visual appearance, and lighting conditions. The policy is given an RGB image of a target kitchen from the new scene and must navigate toward the corresponding target room.

Starting locations are randomly sampled from reachable positions, and the agent approaches the target from different sides. No retraining or scene-specific adaptation is performed.

B Additional Results and analysis

B.1 Target-Room Navigation Accuracy under Visual Variations

Table 7 presents the target-room navigation success rate under progressively stronger visual variations. The agent is given an RGB image of the target kitchen and must reach the corresponding room from randomized starting locations. CEM achieves the highest success rate across all evaluation settings, obtaining 88.45% in the original environment, 86.90% after modifying the target-room semantics, and 84.10% under 80% visual variation. The relatively small decrease of 4.35 percentage points from the original environment to the 80% visual-change setting demonstrates the robustness of the proposed visual episodic memory to changes in scene appearance. In comparison, DNC and TrXL exhibit larger performance drops of 11.43 and 9.79 percentage points, respectively. LSTM also shows a substantial reduction in performance, while SAM remains relatively competitive but consistently below CEM. Under the most challenging 80% visual-change setting, CEM achieves

Table 7: Target-room navigation success rate (%) under increasing visual variations. Higher is better.

Env.	CEM (Ours)	SAM	DNC	TrXL	LSTM
Original	**88.45**	88.01	81.67	75.68	60.25
Target Change	**86.90**	85.29	78.23	70.23	59.23
80% Visual	**84.10**	83.24	70.24	65.89	55.23

Table 8: Navigation path length (mean $\pm$ standard deviation). Lower is better.

Model	Original	Target Change	80% Visual
CEM (Ours)	**14.85 $\pm$ 13.20**	**16.50 $\pm$ 16.10**	**23.80 $\pm$ 24.15**
SAM	17.78 $\pm$ 23.45	18.01 $\pm$ 10.45	24.56 $\pm$ 45.34
DNC	12.45 $\pm$ 10.25	21.23 $\pm$ 23.45	54.67 $\pm$ 55.23
TrXL	43.29 $\pm$ 32.12	45.24 $\pm$ 35.25	55.89 $\pm$ 67.12
LSTM	54.65 $\pm$ 43.54	166.25 $\pm$ 23.45	85.45 $\pm$ 45.23

84.10% success compared with 83.24% for SAM, 70.24% for DNC, 65.89% for TrXL, and 55.23% for LSTM. These results suggest that retrieving representative visual experiences based on their semantic similarity provides a more robust navigation cue than relying solely on recurrent or externally maintained memory representations. Importantly, CEM maintains its target-directed behavior despite substantial changes to the visual evidence encountered during navigation, without requiring explicit position-based memory or scene-specific localization.

B.2 Navigation Path Efficiency

Table 8 reports the navigation path length across the three evaluation settings. CEM consistently produces shorter trajectories than the memory-based and recurrent baselines. In the original environment, CEM achieves a mean path length of 14.85, compared with 17.78 for SAM, 43.29 for TrXL, and 54.65 for LSTM. Under increasing visual changes, path length increases for all methods, but CEM remains more efficient, with a mean path length of 23.80 under the 80% visual-change setting. This indicates that the retrieved visual experiences provide useful navigation cues even when the current observations differ from those encountered during training. Figure 6 compares the number of steps required to reach the goal across different environment variations. CEM achieves the lowest step count in all three settings, requiring 95.30, 100.45, and 118.60 steps in the standard, kitchen-change, and 80% change environments, respectively. SAM exhibits comparable behavior across all three settings, with step counts of 107.25, 110.34, and 123.34. In contrast, the other baselines exhibit substantially larger increases, particularly DNC and LSTM, which require 805.35 and 1009 steps, respectively, under the 80% change. These results demonstrate that CEM reaches the goal efficiently and maintains better robustness as the environment undergoes larger changes, though its margin over SAM narrows relative to the other baselines.

B.3 Navigation with Partial Target Observations

To evaluate the robustness of the proposed visual navigation framework when the target description is incomplete, we introduce a partial target-observation setting. Specifically, 80% of the target image is masked, leaving only a small portion of the visual evidence available to the navigation policy, as illustrated in . The physical target room remains unchanged; only its visual specification is partially occluded. The same partially observed target image is provided to all methods, ensuring a consistent comparison. This setting is particularly challenging because the target embedding is computed from substantially incomplete visual information. Consequently, the agent cannot rely on a direct image-to-image correspondence with the complete target view and must instead exploit the visual experience accumulated during navigation. The experiment therefore evaluates whether episodic visual memory can compensate for missing target information by retrieving previously observed visual states that provide complementary evidence about the target environment. Table 9 reports the results. CEM achieves the highest success rate of 87.20%, while also requiring the fewest navigation steps (210 ± 30) and the shortest path length (21.85 ± 23). In comparison, SAM achieves 86.56% success with 371 ± 42 steps, while DNC, TrXL, and LSTM require substantially more steps to reach lower success rates. The results indicate that the proposed visual episodic memory can maintain

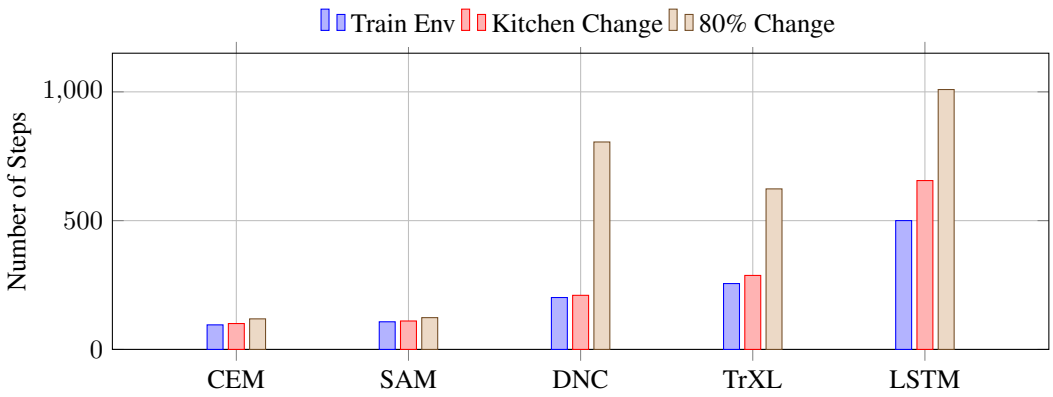


Figure 6: Number of steps required to reach the goal across different environments. Error bars indicate standard deviation. Lower values indicate more efficient goal reaching.

Table 9: Navigation performance with 80% of the target image occluded. Success rate is reported in percentage; path length and navigation steps are reported as mean ± standard deviation. Higher success rate and lower path length/steps are better.

Model	Success Rate (%) ↑	Steps ↓	Path Length ↓
CEM (Ours)	**87.20**	**210 ± 30**	**21.85 ± 23**
SAM	86.56	371 ± 42	22.20 ± 42
DNC	70.00	840 ± 78	52.67 ± 57
TrXL	80.00	646 ± 72	33.25 ± 25
LSTM	65.00	571 ± 30.57	53.85 ± 57

effective target-directed navigation even when 80% of the target visual information is unavailable. Notably, the advantage is not limited to success rate: CEM reaches the target using only 210 ± 30 steps, requiring approximately 1.8×, 4.0×, 3.1×, and 2.7× fewer steps than SAM, DNC, TrXL, and LSTM, respectively.

B.4 Step-Efficiency Analysis on Unseen Environments

Figure 7 reports the number of navigation steps required to reach the target across four unseen environments. CEM requires fewer steps than the compared methods across all environments, while the increase in steps under changing conditions remains moderate. These results provide an additional measure of navigation efficiency alongside the success rate and trajectory length reported in the main paper.

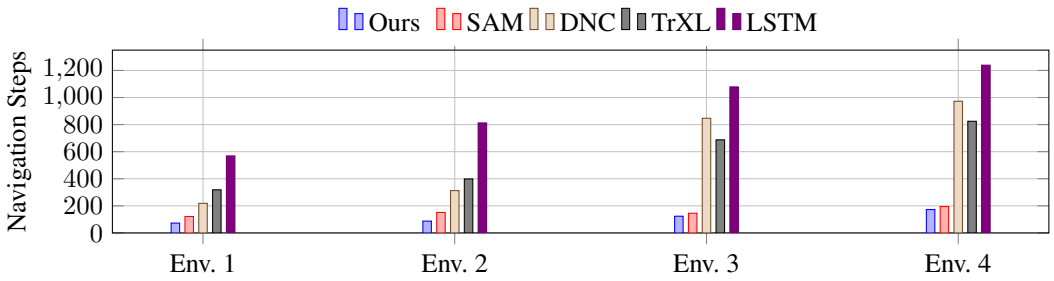


Figure 7: Navigation steps across unseen environments. Bars show the mean number of steps required to reach the target, with error bars representing standard deviation. Lower values indicate more efficient navigation.