NAVTTT: ADAPTIVE PARAMETRIC MEMORY VIA HIERARCHICAL TEST-TIME TRAINING FOR VISION-AND-LANGUAGE NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-and-language navigation (VLN) demands that agents accumulate and reason over streaming visual observations throughout long navigation episodes. Existing approaches struggle to maintain long-term history, constrained by the finite context capacity of transformer architectures. To solve the problem, fast-weight test-time training (TTT) provides a mechanism for encoding streaming observations into an evolving parameter state. However, existing TTT does not distinguish informative observations from repetitive visual content, allowing redundant tokens to interfere with the retention of information needed for subsequent decisions. We propose **Navigation-Conditioned Test-Time Training (NavTTT)**, which introduces adaptive memory formation and hierarchical memory integration for vision-and-language navigation. NavTTT leverages global navigation context to prioritize informative visual evidence and integrates language instructions with spatial information to construct spatially grounded memory representations. Episode-local nonlinear fast weights store these representations as Adaptive Parametric Memory (APM), preserving history beyond the visual context for later decisions. To reduce cross-layer coupling in memory formation, NavTTT maintains independent fast-weight memories across multiple decoder depths. This decoupled design prevents errors or biases from spreading across different memory levels. Meanwhile, it enables shared historical information to be independently represented at various levels of abstraction. Experiments demonstrate strong navigation performance on VLN-CE benchmarks and effective instruction following on a real robot.

1 INTRODUCTION

Vision-and-language navigation (VLN) (Anderson et al., 2018; Krantz et al., 2020) requires an embodied agent to interpret natural-language instructions and sequentially interact with complex visual environments to reach a target destination. Unlike static vision-language understanding, navigation decisions are inherently history-dependent (Hong et al., 2021): the agent must continuously relate its current observation to previously visited locations, encountered landmarks, and completed sub-goals in order to infer its progress and determine the next action (Chen et al., 2021). This makes memory a fundamental component of VLN systems. An effective navigation memory should not only preserve useful historical information over long trajectories, but also selectively retain observations that remain relevant to future decisions (Xu et al., 2026), while avoiding excessive accumulation of repetitive visual content (Zeng et al., 2026).

Existing VLN methods maintain navigation history through either explicit (Chen et al., 2021; 2022; An et al., 2025) or implicit memory (Hong et al., 2021; Krantz et al., 2023; Zeng et al., 2026). Explicit memory organizes past experience into structured representations, such as maps, graphs, or trajectories, making historical state directly accessible to the agent. Implicit memory instead compresses history into latent representations, typically as recurrent features or persistent tokens propagated across time. While less structured, these implicit memories remain fundamentally token-based: retaining more history still requires preserving, updating, or repeatedly attending to an expanding set of latent states. This dependence becomes increasingly problematic in long-horizon navigation, where successive observations are highly redundant and only a small fraction of past

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.

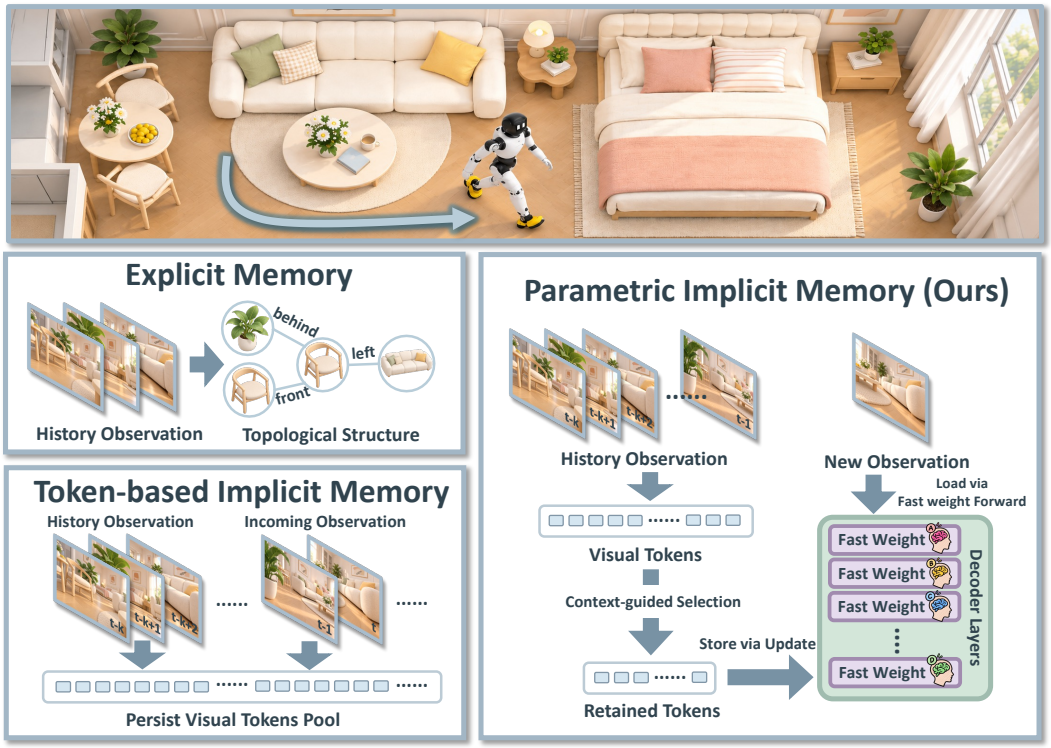

Figure 1: **From explicit to parametric memory in VLN. NavTTT** replaces explicit or token-based history with episode-specific parametric memory stored in fast weights.

information may remain relevant to future decisions. As a result, token-based memory must continuously balance historical coverage against a finite representational and computational budget.

This limitation motivates a different form of implicit memory, where past experience is stored in the model state rather than kept as persistent representations, as illustrated in Fig. 1. Test-time training (TTT) (Sun et al., 2020; 2025) provides a mechanism for doing so by updating fast weights with incoming observations, allowing past information to persist in the model parameters (Behrouz et al., 2025). However, applying TTT directly to VLN is not straightforward. Navigation observations are highly redundant: nearby steps often produce similar visual inputs, while only a few observations remain useful for later decisions. Which observations matter also depends on the instruction, the agent's current location, and its progress along the route. Simply writing every observation into fast weights can therefore fill the memory with repeated or short-lived visual information and make important landmarks harder to preserve. In addition, navigation history needs to support different stages of decision making, while a generic fast-weight memory does not specify how historical information should be used across the navigation model. TTT-based memory for VLN therefore needs to address two key questions: what information should be stored, and how should that information be used during navigation.

To address these challenges, we introduce **Navigation-Conditioned Test-Time Training (NavTTT)**, a parametric memory framework that selectively internalizes navigation experience into episode-specific fast weights. NavTTT conditions memory formation on global navigation context, prioritizing visual evidence relevant to the instruction and navigation state while suppressing redundant content. It further integrates linguistic and spatial cues into the selected evidence, grounding memory in visual appearance and navigation context. To make accumulated experience accessible at multiple stages of navigation reasoning, NavTTT maintains independent fast-weight memories across decoder depths. Each memory transforms the selected evidence through its own association encoders and accumulates it via separate online updates. During subsequent decisions, hidden representations at each depth query the associated memory, integrating historical information with ongoing decoder computation. Together, adaptive memory formation and hierarchical memory in-

tegration allow navigation history to persist beyond limited visual context and inform subsequent decisions through layer-specific encoding and retrieval.

We evaluate NavTTT on the R2R-CE (Anderson et al., 2018) and RxR-CE (Ku et al., 2020) benchmarks to examine the effectiveness of adaptive parametric memory for vision-and-language navigation. NavTTT achieves 66.7% SR and 59.4% SPL on R2R-CE, and 63.6% SR and 52.7% SPL on RxR-CE, consistently outperforming representative VLN baselines. Controlled ablations demonstrate that persistent fast-weight memory, online episode-level adaptation, and adaptive evidence selection each contribute substantially to navigation performance. Together, these results show that selectively internalizing navigation history through fast-weight updates provides an effective mechanism for preserving and exploiting long-horizon experience beyond the active visual context.

- We introduce NavTTT, a parametric memory framework for vision-and-language navigation that internalizes observations into episode-specific fast weights through test-time training, making accumulated experience accessible beyond the explicit visual context.
- We propose navigation-conditioned memory formation that jointly considers task relevance and visual redundancy when selecting evidence, and incorporates linguistic and spatial cues to construct grounded representations for memory updates.
- We develop a hierarchical memory architecture that independently encodes and accumulates shared historical evidence across decoder depths. Layer-specific queries retrieve this information during subsequent decisions, integrating persistent memory into multiple stages of navigation reasoning.

2 Related work

2.1 Vision-Language Navigation and Memory

Vision-language navigation (VLN) has evolved from vision-language pre-training and sequential history modeling toward foundation-model-based navigation. Early approaches leverage large-scale pre-training (Hao et al., 2020) or explicitly model navigation history, such as recurrent state propagation in VLN-BERT (Hong et al., 2021) and long-horizon trajectory encoding in HAMT (Chen et al., 2021). DUET (Chen et al., 2022) further constructs an online topological map to couple global planning with local grounding. More recently, NavGPT (Zhou et al., 2024) and NaviLLM (Zheng et al., 2024) explore LLM-based navigation reasoning, while NaVid (Zhang et al., 2024), Uni-NaVid (Zhang et al., 2025a), and NaVILA (Cheng et al., 2024) extend VLMs toward end-to-end, unified, and real-world navigation. Our work complements this line by focusing specifically on how historical observations are retained and exploited during navigation.

Recent VLN systems have consequently developed increasingly sophisticated memory mechanisms. StreamVLN (Wei et al., 2025) maintains a slow-updating compressed visual context alongside a fast streaming context, while NavMorph (Yao et al., 2025) introduces contextual evolution memory within a self-evolving world model. JanusVLN (Zeng et al., 2026) maintains separate spatial-geometric and visual-semantic histories as compact implicit memories, whereas HiMemVLN (Lyu et al., 2026) organizes navigation history through a hierarchical memory system. Memoir (Xu et al., 2026) further studies retrieval of accumulated environmental and behavioral experience across navigation episodes. Unlike these approaches, NavTTT selectively writes navigation-relevant observations into online-adapted fast weights, treating the model parameters themselves as persistent memory during an episode.

2.2 Parametric Memory and Test-Time Training

Test-time training (TTT) was originally introduced for self-supervised adaptation on unlabeled test samples (Sun et al., 2020), but has recently been repurposed as a sequence-modeling mechanism in which fast weights serve as recurrent state (Sun et al., 2025). This view is closely related to earlier fast-weight formulations of linear attention: Schlag et al. (2021) interpret linear Transformers as fast-weight programmers that store key–value associations in rapidly changing weights. Modern TTT generalizes this perspective by allowing a learnable, potentially nonlinear model to serve as the evolving parameter state. Titans (Behrouz et al., 2025) develops such online neural memory for

long-context modeling, while LaCT (Zhang et al., 2026b) enables large nonlinear fast-weight states through efficient large-chunk updates.

Recent work further explores fast weights as explicit parametric memory for adaptive inference and agents. TMEM (Ren et al., 2026), for example, consolidates interaction experience into lightweight online-updated parameters rather than relying solely on prompt-space memory. NavTTT adopts LaCT's nonlinear fast-weight formulation as its underlying memory operator, but focuses on a different question: how TTT-based memory should be organized for embodied navigation. Specifically, we study *what to write* into fast weights from redundant visual streams and *where to write* this information across the navigation model.

3 METHOD

3.1 PRELIMINARY

Navigation task. Vision-and-language navigation in continuous environments (VLN-CE) tasks an embodied agent with navigating to a target destination by following a natural-language instruction x via sequential interactions with the environment. At each decision step t, the agent observes an egocentric RGB image I_t, maintains a visual history $\mathcal{H}_t$ of up to 7 sampled frames, and receives an episode-relative pose $s_t = (p_t, \sin\psi_t, \cos\psi_t)$, where p_t denotes its 2D position and ψ_t its heading. These inputs are denoted by $\mathcal{X}_t = (x, \mathcal{H}_t, I_t, s_t, t)$. Conditioned on the instruction and all available observations, the navigation policy predicts an action sequence A_t of up to 4 primitive actions drawn from the discrete action space $\mathcal{A} = \{\text{FORWARD}, \text{LEFT}, \text{RIGHT}, \text{STOP}\}$, where t indexes decision cycles rather than individual actions. An episode terminates upon the STOP action or exhaustion of the predefined interaction budget. The core objective is to reach the target location efficiently while faithfully adhering to the given instruction.

Test-Time Training. In the fast-weight formulation, test-time training (TTT) enables online adaptation through a neural memory f_W, whose parameters are updated as new observations arrive, while the pretrained backbone remains fixed during inference. Given a query q_t and selected key–value pairs (k_i, v_i), NavTTT adopts a causal read-before-write order. In the basic gradient-descent form:

$$\begin{aligned} o_t &= f_{W_{t-1}}(q_t), \\ W_t &= W_{t-1} - \eta_t \nabla \mathcal{L}_{\text{mem}}(W_{t-1}), \end{aligned} \tag{1}$$

where W_t denotes the fast weights, η_t is the update rate, and the gradient is taken with respect to W. Following LaCT (Zhang et al., 2026b), the memory objective is the negative dot-product loss:

$$\mathcal{L}_{\text{mem}}(W) = -\sum_i m_i \left\langle f_W(k_i), v_i \right\rangle, \tag{2}$$

where m_i is the binary selection mask and $\langle \cdot, \cdot \rangle$ denotes the inner product. This objective strengthens key–value associations by increasing the agreement between the memory response $f_W(k_i)$ and its associated value v_i. Layer and head indices are omitted; M_t denotes the complete memory state across layers.

3.2 INFORMATIVE TOKEN SELECTION

Task relevance and visual diversity. The relevance of a visual region depends on the navigation instruction and the agent's current state. However, selecting tokens only by relevance may include repeated visual content from the same frame, leaving fewer tokens for other useful regions under a fixed write budget. NavTTT therefore combines context-guided relevance scoring with intra-frame redundancy suppression to select relevant and diverse visual tokens for APM updates. The selection mask controls memory writing, while all visual tokens remain available to the navigation backbone.

Context-Guided Token Selection. Given a visual observation I, raw visual patch features $e_i = \text{sg}\left([E_{\text{vis}}(I)]_i\right)$ are extracted via a frozen vision encoder $E_{\text{vis}}(\cdot)$, and augmented with Fourier positional encodings to generate position-aware visual tokens u_i. In parallel, the Navigation State Encoder (NSE) integrates and encodes non-visual navigation signals, including the navigation instruction g, agent pose s, and decision step k, into a compact explicit state variable c:

$$c = \text{NSE}(g, s, k). \tag{3}$$

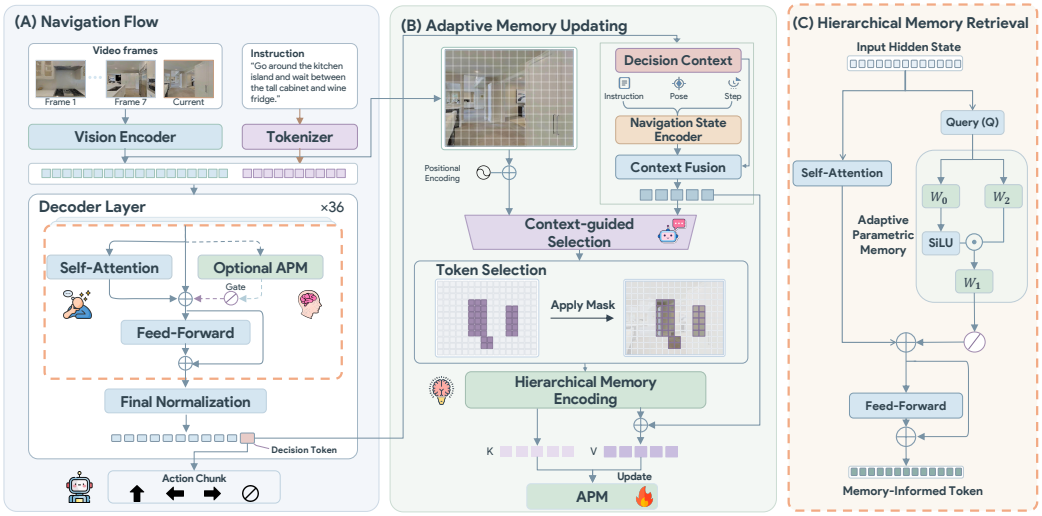

Figure 2: **Overview of NavTTT.** (A) The agent predicts low-level action chunks from language instructions and egocentric visual observations. (B) Navigation-relevant visual tokens are selectively encoded to update hierarchical fast weights online. (C) At subsequent steps, decoder representations query the adapted fast weights across layers, allowing accumulated experience to inform current perception and action prediction.

As illustrated in Fig. 2(B), c is fused with the decoder's high-level decision token d to construct a global guidance context z, expressed as:

$$z = W_d \,\mathrm{RMSNorm}\left(\mathrm{sg}(d)\right) + W_c c, \tag{4}$$

where W_d and W_c denote learnable projection matrices.

This global guidance context serves as a conditional vector that fuses with the features of each visual token along the channel dimension, thereby modulating the feature representation. Based on the accumulated navigation state, it regulates the assessment of visual input relevance, establishing a closed-loop mechanism wherein visual observations are filtered adaptively according to the task state. The relevance score r_i for the i-th token is computed as:

$$r_i = w_s^\top \,\mathrm{SiLU}\left(\mathrm{LayerNorm}\left(W_u u_i + z\right)\right), \tag{5}$$

where W_u is a visual projection matrix and w_s is a learnable scoring vector.

Relying solely on independent scoring fails to suppress visual redundancy, as multiple similar patches may occupy limited memory capacity. To address this, we introduce an incremental diversity selection step. Starting from an empty set, we select tokens one by one, penalizing candidate tokens that are visually similar to those already chosen. See Appendix A.1 for complete details.

3.3 Fast-Weight Memory for Navigation

NavTTT extends the limited visual context capacity by persisting selected task-relevant visual evidence via Adaptive Parametric Memory (APM). Following the LaCT (Zhang et al., 2026b) framework, each memory-augmented decoder layer maintains an independent nonlinear fast-weight mapping to encode accumulated key-value associations. During navigation, this persistent memory complements the self-attention pathway over instructions and current observations, connecting current perception with historical experience from earlier decisions.

3.3.1 Contextual Association Encoding

Given the normalized visual features of each selected region, a layer-specific adapter produces a_i. The key and value representations at decoder layer ℓ are defined as:

$$\begin{aligned} k_i &= \mathcal{T}_K\left(W_K^\ell a_i\right), \\ v_i &= W_V^\ell a_i + W_L^\ell \xi_i + W_z^\ell z, \end{aligned} \tag{6}$$

with the superscript ℓ indicating layer-specific projections, and $\mathcal{T}_K$ applying normalization alongside positional encoding. Specifically, the key encodes visual-positional cues for retrieval, whereas the value aggregates visual appearance a_i, spatial context ξ_i (patch coordinates and relative agent poses), and the fused navigation state z. Ultimately, this formulation grounds the selected visual evidence in its corresponding spatial and task-specific context.

3.3.2 Hierarchical Retrieval and Memory Fusion

NavTTT combines visual context with persistent experience through a hierarchical dual-path decoder. Self-attention is retained throughout the 36-layer backbone, while parallel APM branches are placed at layers $\{5, 11, 17, 23, 29, 35\}$, using zero-based indices. These branches provide memory access at different decoder depths. Each augmented layer maintains independent fast weights, allowing historical evidence to be encoded and retrieved through layer-specific representations.

Each APM branch contains four independent memory heads. Within each head, a query q derived from the decoder hidden state retrieves stored experience through a nonlinear SwiGLU mapping:

$$f_{W^\ell}(q) = W_1^\ell \left[\mathrm{SiLU}(W_0^\ell q) \odot (W_2^\ell q)\right], \tag{7}$$

where $W^\ell = \{W_0^\ell, W_1^\ell, W_2^\ell\}$ denotes the fast weights, and $\odot$ denotes elementwise multiplication. Head indices are omitted for simplicity. This mapping provides access to historical evidence after its source frames leave the active visual context. At decision step t, the self-attention and memory pathways are combined through a residual connection:

$$\widetilde{h}_j = h_j + \mathrm{SelfAttn}(H)_j + \gamma_j \,\mathrm{APM}(h_j; W_{t-1}^\ell), \tag{8}$$

where h_j is the hidden state at position j in the current layer's input sequence H. The APM operator combines the memory-head outputs and projects them into the decoder hidden space. The gate γ_j is fixed to 1 at decision and action positions. For preselected visual tokens in layers $\{11, 17, 23\}$, it is computed from the corresponding normalized pre-decoder visual feature through a learned projection followed by a sigmoid. The gate is set to 0 elsewhere. These visual read positions are determined during prompt processing, before the final decision representation is available. Normalization, headwise scaling, and output projection are included in the corresponding operators. The fused state $\widetilde{h}_j$ is then passed to the standard feed-forward block.

The state W_{t-1}^ℓ contains experience accumulated before decision t and remains unchanged during current prompt processing and action generation. Once the final decision representation is available, the selected key-value pairs from the current observation produce a pending fast-weight update using gradients of the memory association loss $\mathcal{L}_{\mathrm{mem}}$. After the current action chunk is finalized, the updated state is committed as W_t^ℓ and becomes available for retrieval at decision $t + 1$.

4 Experiments

4.1 Experimental Setup

Simulation environments and metrics. Following standard evaluation protocols for continuous vision-and-language navigation (Wei et al., 2025; Zeng et al., 2026), we evaluate NavTTT on the R2R-CE and RxR-CE benchmarks (Krantz et al., 2020; Anderson et al., 2018; Ku et al., 2020). Both benchmarks require agents to follow natural-language instructions and navigate continuous Matterport3D (Chang et al., 2017) environments via low-level actions. Experiments are evaluated on the R2R-CE Val-Unseen split and the RxR-CE Val-Unseen split. R2R-CE performance is reported using NE, OS, SR, and SPL, while RxR-CE performance is reported using NE, SR, SPL, and nDTW. SR and SPL serve as primary metrics for task completion and navigation efficiency.

Real-world evaluation setup. We deploy NavTTT on a humanoid robot (shown in Figure 3) to assess the real-world deployment capability and Sim-to-Real generalization of its learned memory mechanism in continuous navigation tasks. The robot perceives the environment through a forward-facing RGB camera, receiving natural-language instructions alongside streaming egocentric visual observations. NavTTT runs on an off-board inference server equipped with an NVIDIA RTX 4090 GPU, transmitting predicted low-level navigation actions to the robot for execution. No real-world trajectories are used for fine-tuning, enabling a direct assessment of sim-to-real generalization.

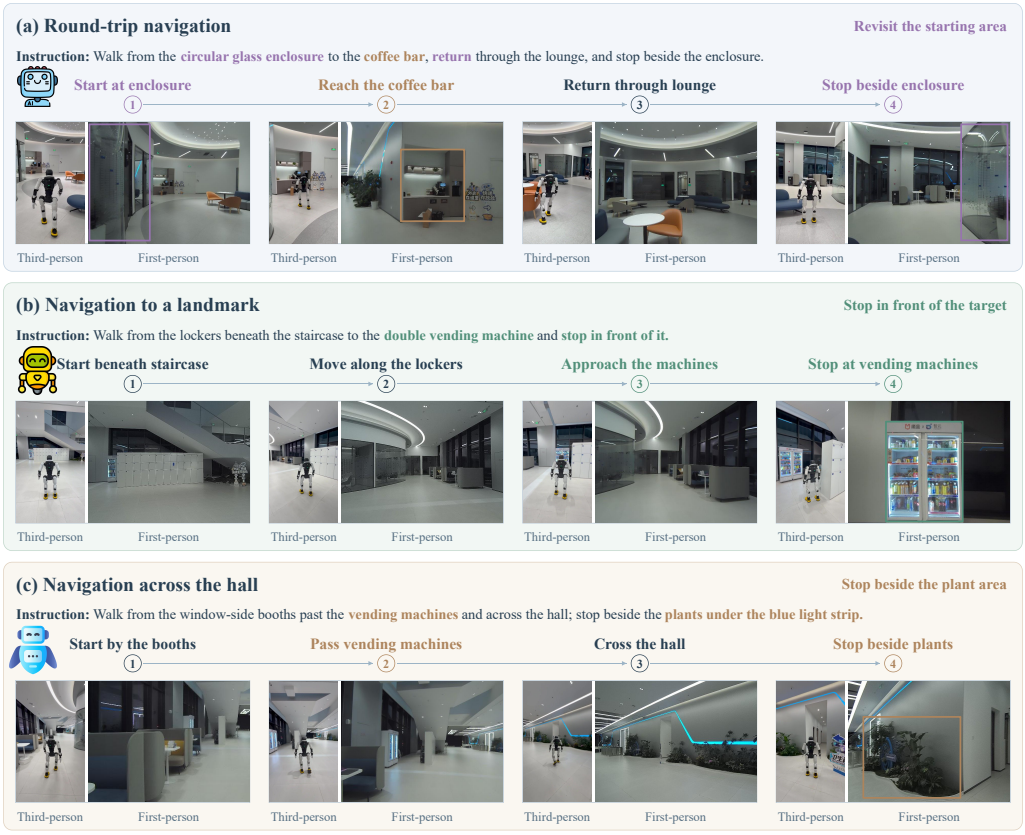

Figure 3: **Real-world qualitative results of NavTTT.** Representative instruction-following episodes on the real robot illustrate NavTTT's navigation behavior in physical environments, guided by natural-language instructions and egocentric RGB observations.

Implementation details. NavTTT is initialized from Qwen3-VL-8B (Bai et al., 2025), with APM integrated from the beginning of joint training. The language backbone, multimodal projector, and APM modules, including the learned fast-weight initialization, are jointly fine-tuned, while the vision encoder remains frozen. Joint training runs for one epoch using AdamW with a weight decay of 10^{-2} and a cosine learning-rate schedule with a warmup ratio of 0.03, consuming approximately 1,910 GPU hours on NVIDIA H20 GPUs. The peak learning rates are 1×10^{-5} for the language backbone and 5×10^{-6} for the multimodal projector. LoRA fine-tuning is used only for the controlled ablation experiments described in Section 4.3. The joint training objective and inference-time fast-weight updates are detailed in Appendix B.

4.2 Main Results

Results on VLN-CE benchmarks. Table 1 compares NavTTT with representative VLN-CE methods across observation modalities and training setups. On R2R-CE Val-Unseen, NavTTT achieves a leading **66.7%** SR and **59.4%** SPL, outperforming JanusVLN by **6.2** SR / **2.6** SPL and StreamVLN by **9.8** SR / **7.5** SPL. These improvements demonstrate how our selective memory pipeline resolves perceptual ambiguities. Context-guided selection isolates task-relevant visual regions, whereas intra-frame redundancy suppression ensures maximal feature diversity within each fast-weight update. NavTTT parametrically registers these distilled historical anchors into fast weights and queries them alongside streaming observations, enabling robust long-horizon contextual reasoning. On RxR-CE Val-Unseen, which presents long-horizon trajectories and detailed instructions, NavTTT reaches **63.6%** SR, **52.7%** SPL, and **71.0%** nDTW. Here, APM retrieves long-range evidence that has exited the active visual window, while the self-attention pathway preserves local grounding, ensuring smooth continuity across extended navigation steps.

Table 1: Comparison among different methods on R2R-CE and RxR-CE Val-Unseen. Pano., Odo., Depth, and S.RGB denote panoramic input, odometry/pose, depth, and a single RGB stream, respectively. Baseline external-data counts follow (Zeng et al., 2026), with $\mathrm{K} = 10^3$ image–action or QA pairs; generic foundation-model pretraining is excluded. "–" denotes an unavailable count or a metric without a verified result for the corresponding split. 0K denotes a reported zero count. NE is measured in meters; other metrics are percentages. Best values are bold.

Method	Observation				R2R-CE Val-Unseen				RxR-CE Val-Unseen				Training
	Pano.	Odo.	Depth	S.RGB	NE↓	OS↑	SR↑	SPL↑	NE↓	SR↑	SPL↑	nDTW↑	External Data
HPN+DN [ICCV21] (Krantz et al., 2021)	✓	✓	✓		6.31	40.0	36.0	34.0	–	–	–	–	–
CMA [CVPR22] (Hong et al., 2022)	✓	✓	✓		6.20	52.0	41.0	36.0	8.76	26.5	22.1	47.0	–
Sim2Sim [ECCV22] (Krantz & Lee, 2022)	✓	✓	✓		6.07	52.0	43.0	36.0	–	–	–	–	–
VLN⬡BERT [CVPR22] (Hong et al., 2022)	✓	✓	✓		5.74	53.0	44.0	39.0	8.98	27.0	22.6	46.7	–
Ego^2-Map [ICCV23] (Hong et al., 2023)	✓	✓	✓		5.54	56.0	47.0	41.0	–	–	–	–	–
DreamWalker [ICCV23] (Wang et al., 2023a)	✓	✓	✓		5.53	59.0	49.0	44.0	–	–	–	–	–
GridMM [ICCV23] (Wang et al., 2023b)	✓	✓	✓		5.11	61.0	49.0	41.0	–	–	–	–	–
InstructNav [CoRL24] (Long et al., 2024)	✓	✓	✓		6.89	–	31.0	24.0	–	–	–	–	–
COSMO [ICCV25] (Zhang et al., 2025b)	✓				–	56.0	47.0	40.0	–	–	–	–	–
AO-Planner [AAAI25] (Chen et al., 2024)	✓		✓		5.55	59.0	47.0	33.0	7.06	43.3	30.5	50.1	–
g3D-LF (Pano.) [CVPR25] (Wang & Lee, 2024)	✓	✓	✓		4.53	68.0	61.0	52.0	–	–	–	–	–
MapNav [ACL25] (Zhang et al., 2026a)		✓	✓	✓	4.93	53.0	39.7	37.2	7.62	32.6	27.7	43.5	–
g3D-LF (Mono.) [CVPR25] (Wang & Lee, 2024)		✓	✓	✓	5.70	59.5	47.2	34.6	–	–	–	–	–
Seq2Seq [ECCV20] (Krantz et al., 2020)			✓	✓	7.77	37.0	25.0	22.0	–	–	–	–	–
NaVid-4D [ICRA25] (Liu et al., 2025)			✓	✓	5.99	55.7	43.8	37.1	–	–	–	–	–
NavMorph [ICCV25] (Yao et al., 2025)			✓	✓	5.75	56.9	47.9	33.2	8.85	30.8	22.8	44.2	–
Sim2Real [CoRL24] (Wang et al., 2024)				✓	5.95	55.8	44.9	30.4	8.79	25.5	18.1	–	0K
NaVid [RSS24] (Zhang et al., 2024)				✓	5.47	49.1	37.4	35.9	8.41	23.8	21.2	–	953K
Uni-NaVid [RSS25] (Zhang et al., 2025a)				✓	5.58	53.3	47.0	42.7	6.24	48.7	40.9	–	3577K
NaVILA [RSS25] (Cheng et al., 2024)				✓	5.22	62.5	54.0	49.0	6.77	49.3	44.0	58.8	13132K
StreamVLN [ICRA26] (Wei et al., 2025)				✓	4.98	64.2	56.9	51.9	6.22	52.9	46.0	61.9	∼ 26330K
JanusVLN [ICLR26] (Zeng et al., 2026)				✓	4.78	65.2	60.5	56.8	6.06	56.2	47.5	62.1	10692K
NavTTT		✓		✓	**3.88**	**73.1**	**66.7**	**59.4**	**4.25**	**63.6**	**52.7**	**71.0**	2974K‡

‡ NavTTT uses 2,974,963 additional DAgger image–action pairs collected from 30,497 trajectories: 780,222 pairs from 10,819 R2R trajectories and 2,194,741 pairs from 19,678 RxR trajectories.

Figure 3 presents representative instruction-following episodes on the real robot. Across 40 real-world navigation trials, NavTTT successfully completes 11, achieving a success rate of **27.5%**. During execution, the agent processes successive egocentric observations and predicts short action chunks for the robot to execute. Its fast-weight memory is updated online with newly observed evidence, making accumulated context available to subsequent decisions. This evaluation complements the simulation benchmarks and provides a quantitative assessment of NavTTT on a physical platform, while highlighting substantial room for improving real-world navigation reliability.

The larger SPL gain over JanusVLN on RxR-CE (5.2 percentage points) than on R2R-CE (2.6 percentage points) is consistent with the motivation to retain historical evidence over longer trajectories. The accompanying improvement in nDTW indicates better alignment with reference routes, complementing the gains in task completion. A possible explanation is that persistent memory makes previously observed landmarks and their navigation context available when recent frames alone provide insufficient information. Context-guided selection and redundancy suppression determine which evidence enters this memory, while hierarchical retrieval allows the retained information to support subsequent decisions. NavTTT obtains these results using approximately 2.97 million additional image–action training pairs. This reported data requirement should be interpreted alongside differences in observation modalities and training settings across methods. Additional qualitative results and failure cases are provided in Appendix C, illustrating successful navigation behavior and situations in which historical memory remains insufficient for reliable action selection.

4.3 Ablation Studies

Following JanusVLN (Zeng et al., 2026), we evaluate component ablations on R2R-CE Val-Unseen. All ablation variants use LoRA fine-tuning and are trained only on the R2R-CE training split, without additional training data. The complete model under this setting is denoted as NavTTT (LoRA), corresponding to A0 in Table 2. All reported performance changes are measured relative to A0.

Table 2: Ablation study on R2R-CE Val-Unseen under the LoRA fine-tuning setting. Bold indicates the best value for each metric.

ID	Variant	NE↓	OS↑	SR↑	SPL↑
A0	NavTTT(LoRA)	4.43	**70.26**	**64.17**	**58.35**
A1	w/o fast-weight memory	4.55	67.75	61.72	55.90
A2	w/o online adaptation (static memory)	4.45	67.65	62.43	55.79
A3	w/o adaptive selection (random-32 writes)	4.50	67.54	61.28	55.77
A4	normalized dense writes	4.40	68.13	60.96	56.09
A5	per-layer hidden-state writer	**4.39**	68.13	60.69	55.36

The results show that the benefit of fast-weight memory depends on both online adaptation and the evidence used for memory formation.

First, online updates provide benefits beyond a static memory branch. Removing fast-weight memory (A1) reduces SR and SPL by 2.45 percentage points each. Retaining the branch while disabling online updates (A2) still produces decreases of 1.74 and 2.56 percentage points, respectively. The A0–A2 comparison is particularly informative because the memory pathway remains available, but its state cannot incorporate newly observed evidence during navigation. The performance gap therefore supports the value of trajectory-dependent memory updates beyond the additional mapping provided by a static branch.

Second, memory quality depends on evidence selection, not simply the number of written tokens. Random selection (A3) preserves the 32-token write budget but reduces SR and SPL by 2.89 and 2.58 percentage points. This comparison links the improvement to the choice of written evidence under the same token budget. Normalized dense writing (A4) also lowers SR and SPL, by 3.21 and 2.26 percentage points, despite including more visual tokens. Together, these results support concentrating updates on a selected set of informative regions within the fixed-capacity memory. They establish the benefit of the combined selection pipeline, while the individual contributions of relevance scoring and redundancy suppression are not isolated by these variants.

Third, the visual writing source matters for effective memory formation. Replacing pre-decoder visual features with per-layer hidden states (A5) produces the largest reductions in SR and SPL, by 3.48 and 2.99 percentage points. A possible explanation is that decoder hidden states already reflect language interactions and previously retrieved memory, making the visual writing source dependent on ongoing decoder computation. The full model anchors appearance features in the current observation and incorporates navigation context through a dedicated encoding pathway. Notably, A3–A5 all obtain lower SR than the memory-free variant A1. Under this setting, adding memory alone does not guarantee an improvement; inappropriate selection or writing representations can offset its potential benefit.

5 Conclusion

This paper presented NavTTT, a framework that uses fast-weight memory to retain historical information for vision-and-language navigation. The method combines navigation-conditioned token selection, contextual association encoding, and memory retrieval at multiple decoder layers. Token selection balances task relevance and visual diversity under a fixed write budget, while contextual encoding associates the selected visual features with spatial information and navigation state. The decoder combines self-attention over the available context with retrieval from episode-local memory, allowing earlier observations to inform subsequent decisions after their source frames leave the active visual window. During inference, the fast weights are updated through a self-supervised objective, while the backbone and other slow-weight parameters remain fixed. Experiments on R2R-CE and RxR-CE show improvements in navigation success and efficiency. We also evaluated the method on a real robot to examine its behavior outside simulation. These results suggest that selectively storing and retrieving historical evidence can complement the limited visual context available to the navigation model. Future work will investigate memory retention over longer trajectories and performance in environments with repeated landmarks or greater visual ambiguity.

AI USE STATEMENT

During the preparation of this work, generative AI tools were used solely to improve readability and language quality, and to assist with software implementation. These tools were not employed for conceiving research ideas or hypotheses, designing experiments, constructing theoretical frameworks, deriving mathematical proofs, analyzing data, or interpreting results. All AI-assisted text and code have been rigorously reviewed and verified by the authors, and the final implementation has passed correctness tests. The authors take full and sole responsibility for the content and integrity of this work.

REFERENCES

Dong An, Hanqing Wang, Wenguan Wang, Zun Wang, Yan Huang, Keji He, and Liang Wang. Etpnav: Evolving topological planning for vision-language navigation in continuous environments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 47(7):5130–5145, 2025. doi: 10.1109/TPAMI.2024.3386695. URL https://doi.org/10.1109/TPAMI.2024.3386695.

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. In *2018 IEEE/CVF conference on computer vision and pattern recognition*, pp. 3674–3683. IEEE, 2018.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/a4ca07aa108036f80cbb5b82285fd4b1-Abstract-Conference.html.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments, 2017. URL https://arxiv.org/abs/1709.06158.

Jiaqi Chen, Bingqian Lin, Xinmin Liu, Lin Ma, Xiaodan Liang, and Kwan-Yee K. Wong. Affordances-oriented planning using foundation models for continuous vision-language navigation, 2024. URL https://arxiv.org/abs/2407.05890.

Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 5834–5847, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/2e5c2cb8d13e8fba78d95211440ba326-Abstract.html.

Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA,*

June 18-24, 2022, pp. 16516–16526. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01604. URL https://doi.org/10.1109/CVPR52688.2022.01604.

An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Xueyan Zou, Jan Kautz, Erdem Biyik, Hongxu Yin, Sifei Liu, and Xiaolong Wang. Navila: Legged robot vision-language-action model for navigation. *CoRR*, abs/2412.04453, 2024. doi: 10.48550/ARXIV.2412.04453. URL https://doi.org/10.48550/arXiv.2412.04453.

Weituo Hao, Chunyuan Li, Xiujun Li, Lawrence Carin, and Jianfeng Gao. Towards learning a generic agent for vision-and-language navigation via pre-training. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 13134–13143. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.01315. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Hao_Towards_Learning_a_Generic_Agent_for_Vision-and-Language_Navigation_via_Pre-Training_CVPR_2020_paper.html.

Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. Vln bert: A recurrent vision-and-language bert for navigation. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 1643–1653, 2021.

Yicong Hong, Zun Wang, Qi Wu, and Stephen Gould. Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation, 2022. URL https://arxiv.org/abs/2203.02764.

Yicong Hong, Yang Zhou, Ruiyi Zhang, Franck Dernoncourt, Trung Bui, Stephen Gould, and Hao Tan. Learning navigational visual representations with semantic map supervision, 2023. URL https://arxiv.org/abs/2307.12335.

Jacob Krantz and Stefan Lee. Sim-2-sim transfer for vision-and-language navigation in continuous environments, 2022. URL https://arxiv.org/abs/2204.09667.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pp. 104–120. Springer, 2020.

Jacob Krantz, Aaron Gokaslan, Dhruv Batra, Stefan Lee, and Oleksandr Maksymets. Waypoint models for instruction-guided navigation in continuous environments, 2021. URL https://arxiv.org/abs/2110.02207.

Jacob Krantz, Shurjo Banerjee, Wang Zhu, Jason Corso, Peter Anderson, Stefan Lee, and Jesse Thomason. Iterative vision-and-language navigation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14921–14930. IEEE, 2023.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding, 2020. URL https://arxiv.org/abs/2010.07954.

Haoran Liu, Weikang Wan, Xiqian Yu, Minghan Li, Jiazhao Zhang, Bo Zhao, Zhibo Chen, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Na vid-4d: Unleashing spatial intelligence in egocentric rgb-d videos for vision-and-language navigation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10607–10615, 2025. doi: 10.1109/ICRA55743.2025.11128467.

Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment, 2024. URL https://arxiv.org/abs/2406.04882.

Kailin Lyu, Kangyi Wu, Pengna Li, Xiuyu Hu, Qingyi Si, Cui Miao, Ning Yang, Zihang Wang, Long Xiao, Lianyu Hu, Jingyuan Sun, and Ce Hao. Himemvln: Enhancing reliability of open-source zero-shot vision-and-language navigation with hierarchical memory system. *CoRR*, abs/2603.14807, 2026. doi: 10.48550/ARXIV.2603.14807. URL https://doi.org/10.48550/arXiv.2603.14807.

Tao Ren, Weiyao Luo, Hui Yang, Rongzhi Zhu, Xiang Huang, Yuchuan Wu, Bingxue Chou, Jieping Ye, Jiafeng Liang, Yongbin Li, et al. Scaling self-evolving agents via parametric memory. *arXiv preprint arXiv:2606.04536*, 2026.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International conference on machine learning*, pp. 9355–9366. PMLR, 2021.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9229–9248. PMLR, 2020. URL http://proceedings.mlr.press/v119/sun20b.html.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. Learning to (learn at test time): Rnns with expressive hidden states. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/sun25h.html.

Hanqing Wang, Wei Liang, Luc Van Gool, and Wenguan Wang. Dreamwalker: Mental planning for continuous vision-language navigation, 2023a. URL https://arxiv.org/abs/2308.07498.

Zihan Wang and Gim Hee Lee. g3d-lf: Generalizable 3d-language feature fields for embodied tasks, 2024. URL https://arxiv.org/abs/2411.17030.

Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. Gridmm: Grid memory map for vision-and-language navigation, 2023b. URL https://arxiv.org/abs/2307.12907.

Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. Sim-to-real transfer via 3d feature fields for vision-and-language navigation, 2024. URL https://arxiv.org/abs/2406.09798.

Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, et al. Streamvln: Streaming vision-and-language navigation via slowfast context modeling. *arXiv preprint arXiv:2507.05240*, 2025.

Yunzhe Xu, Yiyuan Pan, and Zhe Liu. Dream to recall: Imagination-guided experience retrieval for memory-persistent vision-and-language navigation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 48(8):9035–9049, 2026. doi: 10.1109/TPAMI.2026.3679426. URL https://doi.org/10.1109/TPAMI.2026.3679426.

Xuan Yao, Junyu Gao, and Changsheng Xu. Navmorph: A self-evolving world model for vision-and-language navigation in continuous environments. In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pp. 5536–5546. IEEE, 2025. doi: 10.1109/ICCV51701.2025.00525. URL https://doi.org/10.1109/ICCV51701.2025.00525.

Shuang Zeng, Dekang Qi, Xinyuan Chang, Feng Xiong, Shichao Xie, Xiaolong Wu, Shiyi Liang, Mu Xu, and Xing Wei. Janusvln: Decoupling semantics and spatiality with dual implicit memory for vision-language navigation. In *International Conference on Learning Representations*, volume 2026, pp. 33001–33026, 2026.

Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. Navid: Video-based VLM plans the next step for vision-and-language navigation. In Dana Kulic, Gentiane Venture, Kostas E. Bekris, and Enrique Coronado (eds.), *Robotics: Science and Systems XX, Delft, The Netherlands, July 15-19, 2024*, 2024. doi: 10.15607/RSS.2024.XX.079. URL https://doi.org/10.15607/RSS.2024.XX.079.

Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. *Robotics: Science and Systems*, 2025a.

Lingfeng Zhang, Xiaoshuai Hao, Qinwen Xu, Qiang Zhang, Xinyao Zhang, Pengwei Wang, Jing Zhang, Zhongyuan Wang, Shanghang Zhang, and Renjing Xu. Mapnav: A novel memory representation via annotated semantic maps for vision-and-language navigation, 2026a. URL https://arxiv.org/abs/2502.13451.

Siqi Zhang, Yanyuan Qiao, Qunbo Wang, Zike Yan, Qi Wu, Zhihua Wei, and Jing Liu. Cosmo: Combination of selective memorization for low-cost vision-and-language navigation, 2025b. URL https://arxiv.org/abs/2503.24065.

Tianyuan Zhang, Sai Bi, Yicong Hong, Kai Zhang, Fujun Luan, Songlin Yang, Kalyan Sunkavalli, William Freeman, and Hao Tan. Test-time training done right. In *International Conference on Learning Representations*, volume 2026, pp. 157604–157638, 2026b.

Duo Zheng, Shijia Huang, Lin Zhao, Yiwu Zhong, and Liwei Wang. Towards learning a generalist model for embodied navigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pp. 13624–13634. IEEE, 2024. doi: 10.1109/CVPR52733.2024.01293. URL https://doi.org/10.1109/CVPR52733.2024.01293.

Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2024, February 20-27, 2024, Vancouver, Canada*, pp. 7641–7649. AAAI Press, 2024. doi: 10.1609/AAAI.V38I7.28597. URL https://doi.org/10.1609/aaai.v38i7.28597.

A Additional Method Details

NavTTT leverages navigation context to identify key visual regions while retaining selected experiences in fast-weight memory. The following details explain how redundancy is managed during selection and how newly acquired evidence informs subsequent decisions.

A.1 Intra-Frame Redundancy Suppression

The relevance score r_i evaluates each visual token independently. Under a fixed writing budget K, selecting strictly by top relevance scores often causes visual attention to collapse into locally dense regions, packing the memory with semantically duplicate patches. To encourage broad visual coverage, we construct the selection set $\mathcal{S}$ incrementally. Starting from an empty set $\mathcal{S} = \varnothing$, we iteratively select and add one token i^* at a time until reaching capacity K:

$$i^* = \arg\max_{i \notin \mathcal{S}} \left[r_i - \lambda_{\text{div}} \max_{j \in \mathcal{S}} \cos(e_i, e_j) \right], \quad \mathcal{S} \leftarrow \mathcal{S} \cup \{i^*\}. \tag{9}$$

When $\mathcal{S}$ is empty, the redundancy term defaults to zero, ensuring the first token is determined solely by its global relevance r_i. Subsequent tokens dynamically balance high task relevance against similarity to already retained visual evidence, regulated by the diversity coefficient $\lambda_{\text{div}} = 0.25$. Crucially, similarity is measured on raw visual features e_i rather than position-enriched features u_i. This disentangles geometric proximity from feature redundancy, preventing spatial coordinates from confounding true visual appearance overlap and ensuring that selected regions provide genuinely complementary visual information. The finalized set $\mathcal{S}$ determines the binary write mask $m_i = \mathbb{I}[i \in \mathcal{S}]$ for memory updates.

B Training Objectives and Test-Time Training Pipeline

NavTTT is jointly trained for navigation with APM integrated from the outset. At inference, the learned slow-weight parameters remain fixed, while episode-local fast weights are updated online. LoRA is used only in the controlled ablation experiments and is not part of the main training pipeline.

Joint Navigation Training. NavTTT is initialized from Qwen3-VL-8B (Bai et al., 2025), with APM integrated from the beginning of training. The language backbone, multimodal projector, and APM modules, including the token scoring network and learned fast-weight initialization, are jointly optimized under navigation supervision; only the vision encoder remains frozen. This joint training adapts the navigation policy together with the memory-writing and retrieval mechanisms.

Let θ denote these trainable parameters and $\mathcal{D}$ the navigation training set. The training objective is the autoregressive cross-entropy over reference action-token sequences:

$$\mathcal{L}_{\mathrm{nav}}(\theta) = -\mathbb{E}_{\mathcal{D}}\left[\frac{1}{N_t}\log P_\theta(A_t^\star \mid \mathcal{X}_t, M_{t-1})\right], \tag{10}$$

where the expectation averages over training decisions, $A_t^\star$ denotes the reference action chunk, and N_t is its supervised token count. The sequence likelihood factorizes autoregressively over the reference action tokens, conditioned on the navigation inputs $\mathcal{X}_t$ and the memory state M_{t-1} available before the current decision. Prompt context and padding tokens are excluded from the loss.

Test-Time Training via Self-Supervised Fast-Weight Updates. After joint training, all slow-weight parameters remain frozen during inference, including the navigation backbone, multimodal projector, token scoring network, and projections used for memory writing and retrieval. Online adaptation is restricted to the episode-local fast weights W.

At each decision step t, the frozen scoring network computes relevance scores from the current visual features and navigation context. These scores are combined with redundancy suppression to select visual tokens, which are then encoded into contextual key–value pairs. Although the scoring parameters remain fixed, the scores and selected tokens depend on the current inputs.

The selected key–value pairs are used to compute the memory association loss $\mathcal{L}_{\mathrm{mem}}$ defined in Equation 2. Gradients of this loss update only the fast weights, with the keys, values, and write mask held fixed when differentiating with respect to W. The updates incorporate observed evidence into memory without requiring ground-truth action labels.

Following the causal read-before-write order in Equation 1, prompt processing and action generation at step t use M_{t-1}. The fast-weight updates computed from the current observation are committed after the current action chunk is finalized, producing M_t for retrieval at subsequent decisions.

C More Qualitative Results

We present qualitative navigation trajectories from the Habitat simulator to illustrate the performance of NavTTT under complex instructions, along with an analysis of failure cases.

Success and strengths analysis. Figures 4 and 5 highlight successful navigation executions, demonstrating NavTTT's effectiveness across two primary scenarios. First, in multi-step tasks requiring extended traversal, baseline models often suffer from spatial forgetting or looping, whereas NavTTT leverages its fast-weight memory M_t to continuously retain historical context and follow intricate, multi-clause instructions without losing target landmarks. Second, during online execution, self-supervised updates driven by $\mathcal{L}_{\mathrm{mem}}$ enable the fast-weight memory to dynamically adapt to unseen visual features, ensuring smooth and accurate trajectories.

Failure case analysis. Figure 6 illustrates three representative failure modes. First, directional drift or reversal leads the agent away from the instructed route, highlighting difficulties in maintaining directional consistency during navigation. Second, repetitive action loops produce little progress toward the goal, suggesting a need for better detection of unproductive behavior and more effective recovery strategies. Third, missed turn-offs occur when the agent passes a required branching

point without executing the instructed turn, highlighting the importance of coordinating local visual grounding with action timing. Together, these cases reveal remaining challenges in translating perceptual and historical context into reliable local decisions. Improved directional grounding, progress-aware recovery, and timely landmark-conditioned turning are promising directions for addressing these limitations.



Figure 4: Qualitative results of NavTTT on RxR-CE.



Figure 5: Qualitative results of NavTTT on R2R-CE.



Figure 6: Visualization and presentation of the types of failure cases.