SC^2-WM: A Self-Correcting World Model with Closed-Loop Feedback for Vision-and-Language Navigation in Continuous Environments

Anonymous Authors[1]

Abstract

Vision-and-Language Navigation in Continuous Environments (VLN-CE) requires agents to make fine-grained navigation decisions under partial observability. However, most existing methods rely on open-loop execution, lacking mechanisms to detect and correct internal state drift during inference. We propose SC^2-WM, a self-correcting world model framework that introduces internal feedback for closed-loop decision making in VLN-CE. Our method derives feedback from world-model foresight to perform state-level plan refinement before action execution. To handle challenging scenarios, we further introduce conditional world-aware adaptation, which enables model-level correction by selectively updating the world model at test time when feedback indicates model capacity insufficiency. Experiments on standard VLN-CE benchmarks demonstrate improved navigation robustness and generalization. Code is available in the Supplementary Material.

1. Introduction

Vision-and-Language Navigation (VLN) (Anderson et al., 2018b; Qi et al., 2020; Chen et al., 2022c) is a representative embodied AI task that requires an agent to navigate autonomously in 3D environments following natural language instructions. Targeting real-world interactive settings, this task captures tight coupling between language understanding and spatial decision making, making it a central topic in multimodal and embodied intelligence research (Zhang et al., 2024; Gao et al., 2025; Yuan et al., 2025).

Extending Vision-and-Language Navigation to continuous environments (VLN-CE) (Krantz et al., 2020; An et al., 2024) imposes substantially stricter demands on embodied agents. Beyond fine-grained control in a continuous action space, the agent must continuously update its understanding of the environment during execution. Moreover, real-world navigation is inherently partially observable, requiring the agent to integrate historical information into an internal representation while remaining aligned with the semantic constraints of natural language instructions.

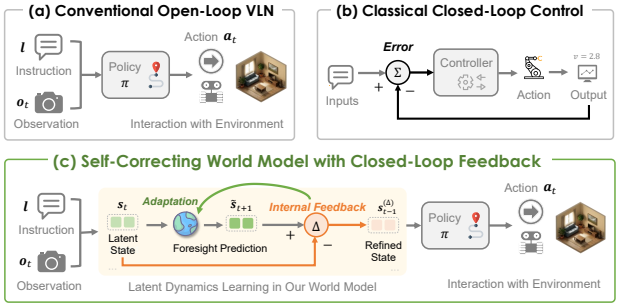

Figure 1. Comparison of (a) conventional open-loop VLN framework, (b) classical closed-loop control, and (c) our self-correcting world model with closed-loop feedback. Unlike open-loop methods that execute actions without validation, SC^2-WM leverages internal feedback derived from foresight prediction to refine latent states before action execution.

Despite recent advances in model architectures and decision policies for VLN-CE (An et al., 2024; Wang et al., 2024b), most existing approaches still adopt an open-loop execution paradigm at inference time, as shown in Figure 1(a). The agent predicts actions from current observations and history, and executes them without validating their consequences. As a result, early decision errors may accumulate over time, degrading navigation performance. Classical closed-loop control (Figure 1(b)) addresses this by comparing outputs against references to compute error signals. While introducing closed-loop feedback mechanisms seems a natural solution (Bu et al., 2024; Li et al., 2024), doing so in VLN-CE is inherently challenging: external supervision is typically sparse and delayed, with success only observable at the trajectory level. Prior attempts incorporate feedback through reinforcement learning (Wang et al., 2024a) or model predictive control (Dey & Bhasin, 2025), but often rely on manually designed rewards and suffer from training instability. Test-time adaptation methods (Wang

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

et al., 2020; Gao et al., 2024) instead adjust models using output-level statistics such as uncertainty or entropy, yet these signals remain decoupled from the agent's internal representation and cannot directly reveal misalignment in its understanding of the environment.

Based on the above analysis, we argue that effective closed-loop decision making in VLN-CE hinges on constructing appropriate feedback signals. Since external rewards are difficult to obtain or design reliably, we turn to an alternative: 'can the agent extract internal feedback signals from its own decision process for self-correction?' Predictive processing in cognitive science (Huang & Rao, 2011) suggests that adaptive behavior relies on continuously monitoring the consistency between internal predictions and subsequent observations, rather than on single-pass inference or delayed external supervision. This perspective aligns well with VLN-CE, where navigation failures often arise not from a single erroneous observation, but from gradual drift in the agent's internal understanding of the environment or instruction. If such drift can be detected during execution, the agent can correct its representation before errors amplify further.

Inspired by this, we propose SC^2-WM, a self-correcting world model framework for VLN-CE that enables closed-loop decision making through a computable internal feedback signal (Figure 1(c)). SC^2-WM focuses on detecting and correcting deviations in the agent's internal understanding of the environment during navigation. The core of SC^2-WM is a world model that captures environment dynamics and produces foresight predictions from latent state representations. These predictions reflect both the anticipated evolution of the environment and the agent's current internal assumptions. During inference, the discrepancy between the current latent state and its foresight provides an internal reference signal indicating how the selected action would steer the agent's internal dynamics before it is executed.

Based on this signal, SC^2-WM introduces a dual-level self-correction mechanism. *(i) At the state level*, feedback-guided plan refinement performs state-level correction by modulating the current latent state using foresight-derived signals, mitigating local inference drift during action selection. *(ii) At the model level*, conditional world-aware adaptation targets model-level correction by updating the world model when feedback reveals heavy reliance on foresight-derived guidance, improving its generalization ability for the testing environments. Together, these mechanisms enable effective navigation in continuous, dynamic environments.

Our main contributions are summarized as follows:

- We propose SC^2-WM, a dual-level self-correcting world model framework for VLN-CE that leverages internal foresight to enable closed-loop decision making with internal generated feedback.
- We introduce a dual-level self-correction mechanism, comprising feedback-guided plan refinement for immediate state-level correction and conditional world-aware adaptation for targeted model-level correction at test time.
- We conduct extensive experiments on standard VLN-CE benchmarks and real-world deployment, demonstrating that SC^2-WM consistently improves navigation performance.

2. Related Work

VLN-CE. Vision-and-Language Navigation (VLN) has emerged as a cornerstone of embodied AI, requiring agents to navigate unseen environments following natural language instructions (Anderson et al., 2018b; Qi et al., 2020; Banerjee et al., 2021; Chen et al., 2022b; An et al., 2022a; Qiao et al., 2023; Song et al., 2025). While early works operated in discrete settings, Krantz et al. (2020) introduced VLN-CE to narrow the sim-to-real gap. This setting transitions agents from graph-based teleportation to continuous 3D spaces, introducing higher demands for understanding the environment state. To address the partial observability and lack of topological priors in VLN-CE, many approaches predominantly relied on lightweight navigation models, enhancing agents' decision-making and execution stability in complex environments through improved state representations (Wang & Lee, 2025; Huang et al., 2025), trajectory modeling (An et al., 2024; Wang et al., 2023c), and action selection (Wang et al., 2025c; 2024b). Subsequently, with the rapid advancement of multimodal large-language models (MLLMs), recent works (Lin et al., 2025; Zhang et al., 2025; 2024; Gao et al., 2025; Yuan et al., 2025) have explored incorporating MLLMs into VLN to leverage their strong language understanding and cross-modal reasoning abilities, thereby enhancing the modeling of complex instructions, long-term dependencies, and high-level semantic relationships. Despite their promising semantic performance, such approaches typically incur substantial computational overhead and still face limitations in online inference efficiency and edge deployment. Therefore, We focus on the lightweight model paradigm for VLN-CE, investigating how to systematically improve navigation decision-making and online adaptability within a compact model framework, without focusing on large-scale MLLMs.

World Models. World models are fundamentally defined as internal representations that capture environmental dynamics, enabling agents to simulate future states based on current observations and potential actions (Ha & Schmidhuber, 2018; Hafner et al., 2025). This predictive capability has been extensively applied across various embodied tasks, ranging from game simulation in virtual environments (Valevski et al., 2024) to autonomous driving (Ren et al.,

2025; Russell et al., 2025) and robotic manipulation (Chi et al., 2025; 2024). In the context of VLN, World models empower agents to overcome partial observability through predictive foresight. DreamWalker (Wang et al., 2023b) integrates scene synthesis with Monte Carlo Tree Search (MCTS) for decision making. NavMorph (Yao et al., 2025) employs a recurrent state-space model to characterize the evolution of internal states. However, these approaches follow an open-loop paradigm, in which action execution is not guided by feedback signals, making them prone to the accumulation of navigation errors. Although a few works (Huang et al., 2025) explore predictive feedback, they are limited to modeling discrete environments and do not explicitly quantify error signals for joint correction at both the decision and model levels. To date, how to construct closed-loop, feedback-driven world models for tasks with sparse and delayed supervision remains an open problem.

Closed-Loop Mechanisms for Embodied Tasks. The goal of closed-loop control is to continuously adjust actions based on real-time sensory feedback to minimize the discrepancy between the current state and the target state (Hutchinson et al., 2002; Levine et al., 2016). Early approaches typically focused on tasks with real-time error sensors, such as visual servoing (Hutchinson et al., 2002). However, in contemporary embodied manipulation or navigation tasks, the absence of supervision signals often makes it difficult to compute error signals in real-time, hindering the construction of closed-loop systems. To address this, CLOVER (Bu et al., 2024) uses video diffusion models to generate visual sub-goals for establishing closed-loop control during manipulation. The fast-slow system paradigm (Li et al., 2024) leverages the semantic reasoning capability of large language models to construct feedback signals, where the fast system is responsible for efficient execution and the slow system is used for failure reflection and correction. Other methods have explored the design of feedback signals in VLA models, delegating closed-loop control to lightweight policies (Sendai et al., 2025). In recent years, world models have become a viable approach for building closed-loop feedback mechanisms due to their ability to predict future states. However, current explorations in this area are primarily based on the Model Predictive Control (MPC) framework (Sendai et al., 2025), which heavily relies on reinforcement learning and is difficult to train. Therefore, this paper focuses on exploring how to efficiently extract internal feedback signals from the world model's own decision-making process to enable self-correction at both the decision and model levels.

3. Our Approach

In this section, we propose SC^2-WM, a self-correcting world model framework for VLN-CE. Our model incorporates an internal feedback mechanism to support closed-loop decision making, enabling self-correction to enhance navigation.

Task Definition. In VLN-CE (Krantz et al., 2020; Krantz & Lee, 2022; Wang et al., 2025c), an agent navigates in continuous 3D environments given RGB-D observations and a natural language instruction. Each episode starts from an initial position and terminates when the agent issues a 'STOP' action or reaches a predefined maximum number of steps. At each timestep t, the agent receives an observation $\boldsymbol{o}_t$ and selects a navigation action a_t based on predicted candidate waypoints (Krantz et al., 2021; Hong et al., 2022), which is then executed via low-level continuous control. Formally, the agent follows a learnable policy π that maps the instruction $\boldsymbol{l}$, observation history $\boldsymbol{o}_{1:t}$, and past actions $\boldsymbol{a}_{1:t-1}$ to the current action: $\boldsymbol{a}_t \sim \pi(\boldsymbol{a}_t \mid \boldsymbol{l}, \boldsymbol{o}_{1:t}, \boldsymbol{a}_{1:t-1})$.

Framework Overview. As illustrated in Figure 2, we propose SC^2-WM, a self-correcting world model framework that integrates internal foresight with closed-loop feedback for robust decision making in VLN-CE. The framework supports two complementary forms of self-correction: *(i) feedback-guided plan refinement*, which performs immediate, state-level correction before action execution; and *(ii) conditional world-aware adaptation*, which selectively updates the world model at test time when feedback signals challenging scenarios, achieving model-level correction.

At each timestep t, the agent encodes the current observation $\boldsymbol{o}_t$ into visual features $\boldsymbol{v}_t$ and integrates them with the linguistic representation extracted from the instruction $\boldsymbol{l}$ (Chen et al., 2022c; An et al., 2024). These features are then incorporated into a latent state $\boldsymbol{s}_t$, maintained by the world model to capture the navigational context. This state is updated recurrently via a posterior transition $q(\boldsymbol{s}_t \mid \cdot)$ that integrates the previous state, the executed action, and the current visual-linguistic features. Based on $\boldsymbol{s}_t$, a foresight policy $\tilde{\pi}$ proposes a provisional action $\tilde{\boldsymbol{a}}_t$, and the world model anticipates a foresight state $\tilde{\boldsymbol{s}}_{t+1}$ via a prior transition $p(\tilde{\boldsymbol{s}}_{t+1} \mid \cdot)$ without observing new visual input. An internal feedback signal derived from this foresight is used for plan refinement, correcting the current state to $\boldsymbol{s}_t^{(\Delta)}$ (detailed in Section 3.1), from which the navigation policy π selects the final action $\boldsymbol{a}_t$. We formalize SC^2-WM as follows:

$$\text{Visual Representation: } \boldsymbol{v}_t = f_{\text{enc}}(\boldsymbol{o}_t), \tag{1}$$

$$\text{Initial Latent State: } \boldsymbol{s}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{2}$$

$$\text{Posterior State Update: } \boldsymbol{s}_t \sim q\big(\boldsymbol{s}_t \mid \boldsymbol{s}_{t-1}^{(\Delta)}, \boldsymbol{a}_{t-1}, \boldsymbol{v}_t, \boldsymbol{l}\big), \tag{3}$$

$$\text{Provisional Decision: } \tilde{\boldsymbol{a}}_t \sim \tilde{\pi}(\cdot \mid \boldsymbol{s}_t), \tag{4}$$

$$\text{Prior State Transition: } \tilde{\boldsymbol{s}}_{t+1} \sim p\big(\tilde{\boldsymbol{s}}_{t+1} \mid \boldsymbol{s}_t, \tilde{\boldsymbol{a}}_t\big), \tag{5}$$

$$\text{Action Prediction: } \boldsymbol{a}_t \sim \pi(\cdot \mid \boldsymbol{s}_t^{(\Delta)}), \tag{6}$$

$$\text{Visual Decoder: } \tilde{\boldsymbol{v}}_{t+1} \sim p\big(\tilde{\boldsymbol{v}}_{t+1} \mid \tilde{\boldsymbol{s}}_{t+1}\big), \tag{7}$$

Here, the visual decoder reconstructs future visual features

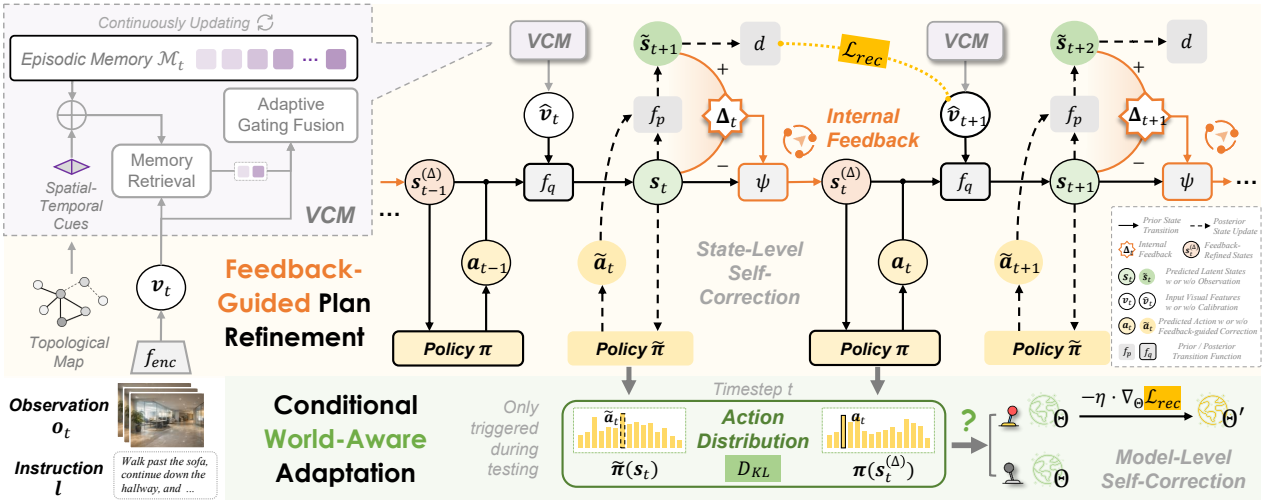

Figure 2. **Overview of SC^2-WM framework,** which has two complementary self-correction modules: (i) feedback-guided plan refinement for state-level correction before action execution, and (ii) conditional world-aware adaptation for model-level correction during test time.

from the foresight state, providing a learning signal for training the world model, as described in Section 3.2.

When the above feedback-guided plan refinement leads to a pronounced change in the action distributions generated by $\tilde{\pi}(\boldsymbol{s}_t)$ and $\pi(\boldsymbol{s}_t^{(\Delta)})$, it suggests that foresight-derived guidance plays a substantial role in shaping the current decision. In such cases, SC^2-WM selectively activates conditional world-aware adaptation to further enhance its modeling capacity for the current environment dynamics. This adaptation operates at the model level by updating the world model parameters using its internal self-supervised objective. We detail the adaptation procedure in Section 3.3.

3.1. Feedback-Guided Plan Refinement

Having introduced the overall pipeline, this section focuses on how the internal feedback signal is constructed and used for plan refinement. In our latent-based world model, an effective feedback mechanism critically depends on the quality of the latent state representation. To serve as a reliable basis for feedback, the latent state should capture action-relevant dynamics while suppressing variations from viewpoint changes, partial observability, and rendering noise. Otherwise, the state shift induced by foresight may become ambiguous, making it unclear whether the predicted change reflects the agent's decision or incidental perceptual fluctuations. To mitigate this, we first calibrate the visual representation so that the latent state more faithfully reflects action-conditioned dynamics.

Visual Calibration Module (VCM). VCM maintains an episodic memory of recent visual representations, $\mathcal{M}_t = \{(\boldsymbol{k}_i, \boldsymbol{m}_i)\}_{i=t-L}^{t-1}$, where L denotes the memory horizon. Each memory element $\boldsymbol{m}_i$ stores visual features extracted from past observations. To retrieve relevant context for calibration, VCM constructs attention keys on the fly by augmenting the stored visual features with relative spatial and temporal cues with respect to the current step t (see Appendix for details).

Given the current visual representation $\boldsymbol{v}_t$ as the query, VCM retrieves contextual information from $\mathcal{M}_t$ via cross-attention with learnable temporal decay:

$$\boldsymbol{c}_t = \sum_i \mathrm{softmax}(A_i)\, \boldsymbol{m}_i, \quad A_i \propto \boldsymbol{v}_t^\top \boldsymbol{k}_i - \varphi(\delta t_i), \quad (8)$$

where $\delta t_i = t - i$ is the temporal distance and $\varphi(\cdot)$ down-weights temporally distant observations.

The retrieved context $\boldsymbol{c}_t$ is then fused with the current representation via adaptive gating: $\hat{\boldsymbol{v}}_t = \boldsymbol{\alpha}_t \odot \boldsymbol{v}_t + (1 - \boldsymbol{\alpha}_t) \odot \boldsymbol{c}_t$, where $\boldsymbol{\alpha}_t$ is a learnable weight.

By aggregating observations from multiple viewpoints along the trajectory, the calibrated representation $\hat{\boldsymbol{v}}_t$ captures richer geometric and semantic cues about the environment, effectively mitigating ambiguity arising from partial observability and perceptual noise (Arandjelovic et al., 2016). This multi-view integration provides a more stable basis for latent-state inference, ensuring that state shifts induced by foresight primarily reflect action-relevant dynamics rather than incidental perceptual fluctuations.

Internal Feedback. With the calibrated visual input providing a stable latent state $\boldsymbol{s}_t$, which better captures action-conditioned dynamics, we now construct an internal feedback signal from foresight to guide plan refinement. Based on $\boldsymbol{s}_t$, the foresight policy $\tilde{\pi}$ proposes a provisional action $\tilde{\boldsymbol{a}}_t$. Since future observations are unavailable before execution, we leverage the world model's prior transition f_p

to anticipate the outcome: $\tilde{\boldsymbol{s}}_{t+1} = f_p(\boldsymbol{s}_t, \tilde{\boldsymbol{a}}_t)$. Here, f_p is implemented as a Transformer-based transition model. Note that while the world model formulation in Eq.(5) models the prior transition as a distribution, we use the deterministic mean prediction (Min et al., 2024) to obtain $\tilde{\boldsymbol{s}}_{t+1}$, avoiding the introduction of additional noise into the feedback signal.

The foresight state $\tilde{\boldsymbol{s}}_{t+1}$ encodes the anticipated consequence of the provisional action. However, it represents a future latent state rather than actionable information for the current decision. We therefore extract the action-induced change by computing the latent discrepancy:

$$\boldsymbol{\Delta}_t = \tilde{\boldsymbol{s}}_{t+1} - \boldsymbol{s}_t. \tag{9}$$

Notably, $\boldsymbol{\Delta}_t$ is not a training target to be minimized, but a model-internal guidance signal that captures how the provisional action would steer the latent dynamics.

Based on this guidance, we perform self-correction by updating the current latent:

$$\boldsymbol{s}_t^{(\Delta)} = \psi(\boldsymbol{s}_t, \boldsymbol{\Delta}_t), \tag{10}$$

where $\psi(\cdot)$ is a learnable neural network and $\boldsymbol{s}_t^{(\Delta)}$ is the feedback-refined state for final action prediction (Eq.(6)).

3.2. Pre-training Objective

The pre-training of SC^2-WM aims to learn an internal latent state $\boldsymbol{s}_t$ that is suitable for feedback-guided plan refinement. Specifically, the learned state is expected to (i) support reliable action prediction and (ii) encode predictive latent dynamics that anticipate future observations. Together, these properties provide the necessary foundation for feedback-guided refinement mechanism introduced in Section 3.1 and conditional world-aware adaptation described in Section 3.3.

We first impose action-level supervision on both original latent state $\boldsymbol{s}_t$ and feedback-refined latent state $\boldsymbol{s}_t^{(\Delta)}$, ensuring a consistent decision basis before and after refinement:

$$\mathcal{L}_{\text{ac}} = -\,\mathbb{E}_t\Big[\log p(a_t^* \mid \boldsymbol{s}_t) + \log p(a_t^* \mid \boldsymbol{s}_t^{(\Delta)})\Big], \tag{11}$$

where a_t^* denotes the expert action at time step t. In practice, this objective is implemented using standard imitation learning supervision (An et al., 2024; Wang et al., 2025c).

In addition, we regularize the prior transition (Eq.(5)) by aligning it with the observation-conditioned posterior (Eq.(3)) via KL divergence, encouraging distributional consistency between inference and foresight:

$$\mathcal{L}_{\text{wm}} = \mathbb{E}_t\Big[D_{\text{KL}}\big(q(\boldsymbol{s}_t \mid \cdot) \;\big\|\; p(\tilde{\boldsymbol{s}}_t \mid \cdot)\big)\Big], \tag{12}$$

However, distributional alignment alone does not guarantee that the foresight state captures meaningful environment dynamics. To this end, we further supervise the action-conditioned prior transition by its ability to anticipate future visual features. Specifically, given the foresight state $\tilde{\boldsymbol{s}}_{t+1}$ predicted by the world model, a lightweight decoder $d(\cdot)$ maps it to an estimate of the next-step front-view visual representation (Eq.(7)). The supervision signal $\boldsymbol{v}_{t+1}$ is obtained by encoding the observation at $t{+}1$ with a pretrained visual encoder (Dosovitskiy, 2020):

$$\mathcal{L}_{\text{rec}} = \mathbb{E}_t\Big[\,\|d(\tilde{\boldsymbol{s}}_{t+1}) - \boldsymbol{v}_{t+1}\|_2^2\Big]. \tag{13}$$

Although defined in the visual feature space, this objective directly constrains the transitioned latent state, encouraging the world model to encode task-relevant environment dynamics without resorting to pixel-level reconstruction.

The final training objective jointly optimizes action prediction and predictive dynamics by combining navigation and world model losses: $\mathcal{L} = \mathcal{L}_{\text{ac}} + \mathcal{L}_{\text{wm}} + \mathcal{L}_{\text{rec}}$.

3.3. Conditional World-Aware Adaptation

The feedback-guided refinement enables immediate self-correction by modulating the latent state before action execution, leveraging foresight from the trained world model. Its effectiveness thus depends on how accurately the world model captures the current environment dynamics.

During deployment, the agent may encounter environments whose visual appearance or dynamics differ from training. Under such domain shifts, the prior transition (Eq.(5)) learned offline may become less aligned with the current observations, reducing the informativeness of the resulting feedback signal. Adapting the world model to better reflect environment-specific characteristics can improve foresight quality, thereby strengthening feedback-guided refinement. We therefore introduce a conditional world-aware adaptation mechanism that selectively updates the world model parameters at test time, activated only when the agent's decision relies heavily on feedback guidance.

To determine when adaptation is beneficial, we monitor the magnitude of feedback-induced corrections via the KL divergence between action distributions before and after refinement: $\kappa_t = D_{\text{KL}}(\tilde{\boldsymbol{p}}_t \,\|\, \boldsymbol{p}_t)$, where $\tilde{\boldsymbol{p}}_t = \text{softmax}(\tilde{\pi}(\boldsymbol{s}_t))$ and $\boldsymbol{p}_t = \text{softmax}(\pi(\boldsymbol{s}_t^{(\Delta)}))$. A large κ_t indicates that feedback has substantially altered the agent's action preference, suggesting that the world model would benefit from adapting to the current environment dynamics.

Instead of relying on a fixed threshold, we maintain a running buffer of recent κ values and trigger adaptation when κ_t exceeds the τ-th percentile of this empirical distribution. We further require that the entropy of $\boldsymbol{p}_t$ falls below a threshold ϵ, ensuring that updates occur only when the corrective signal is both substantial and confident.

Table 1. Experimental results on R2R-CE dataset. Results better than **base model** are shown in blue. Best results for both panoramic and monocular settings are underlined. * indicates experimental results that we have reproduced.

Camera	Methods	Val Seen					Val Unseen					Test Unseen				
		TL ↓	NE ↓	OSR	SR	SPL	TL ↓	NE ↓	OSR	SR	SPL	TL ↓	NE ↓	OSR	SR	SPL
Monocular	LAW (Raychaudhuri et al., 2021) [EMNLP21]	9.34	6.35	49	40	37	8.89	6.83	44	35	31	9.67	7.69	28	38	25
	CM^2 (Georgakis et al., 2022) [CVPR22]	12.05	6.10	50.7	42.9	34.8	11.54	7.02	41.5	34.3	27.6	13.90	7.70	39	31	24
	WS-MGMap (Chen et al., 2022a) [NeurIPS22]	10.12	5.65	51.7	46.9	43.4	10.00	6.28	47.6	38.9	34.3	12.30	7.11	45	35	28
	NaVid (Zhang et al., 2024) [RSS24]	-	-	-	-	-	-	5.47	49.1	37.4	35.9	-	-	-	-	-
	ETPNav/p (Wang et al., 2025c) [CoRL24]	-	-	-	-	-	-	6.81	42.4	32.9	23.1	-	-	-	-	-
	NavMorph (Yao et al., 2025) [ICCV2025]	20.03	4.58	62.7	55.8	38.9	22.54	5.75	56.9	47.9	33.2	24.75	6.01	54.5	45.7	30.2
	g3D-LF (Wang & Lee, 2025) [CVPR2025]	-	-	-	-	-	-	5.70	59.5	47.2	34.6	-	6.00	57.5	46.3	32.2
	VLN-3DFF (Wang et al., 2025c) [CoRL24]	-	-	-	-	-	-	5.95	55.8	44.9	30.4	-	6.24	54.4	43.7	28.9
	VLN-3DFF*	22.90	4.92	62.1	52.7	36.7	26.16	6.05	54.9	43.8	29.4	26.02	6.22	54.7	43.8	28.6
	SC^2-WM	17.26	4.53	64.3	56.0	41.9	17.65	5.37	58.8	50.9	37.2	21.68	6.04	57.1	47.0	32.1
Panoramic	Seq2Seq (Anderson et al., 2018b) [CVPR18]	9.26	7.12	46	37	35	8.64	7.37	40	32	30	8.85	7.91	36	28	25
	Sim2Sim (Krantz & Lee, 2022) [ECCV22]	11.18	4.67	61	52	44	10.69	6.07	52	43	36	11.43	6.17	52	44	37
	CWP-CMA (Hong et al., 2022) [CVPR22]	11.47	5.20	61	51	45	10.90	6.20	52	41	36	11.85	6.30	49	38	33
	CWP-BERT (Hong et al., 2022)[CVPR22]	12.50	5.02	59	50	44	12.23	5.74	53	44	39	13.51	5.89	51	42	36
	DREAMW (Wang et al., 2023a) [ICCV23]	11.60	4.09	59	66	48	11.30	5.53	49	59	44	11.80	5.48	49	57	44
	GridMM (Wang et al., 2023c) [ICCV23]	12.69	4.21	69	59	51	13.36	5.11	61	49	41	13.31	5.64	56	46	39
	BEVBert (An et al., 2022a) [ICCV23]	13.98	3.77	73	68	60	13.27	4.57	67	59	50	15.31	4.70	67	59	50
	FSTTA (Gao et al., 2024) [ICML24]	12.39	4.25	69	58	50	11.58	5.27	58	48	42	13.17	5.84	55	46	38
	Dynam3D (Wang et al., 2025b) [ICCV2025]	-	-	-	-	-	-	5.34	62	53	46	-	5.53	60	51	45
	NavMorph (Yao et al., 2025) [ICCV2025]	11.76	3.66	78	70	62	12.68	4.37	68	64	53	12.69	4.69	68	60	52
	g3D-LF (Wang & Lee, 2025) [CVPR2025]	-	-	-	-	-	-	4.53	68	61	52	-	4.78	68	58	51
	ETPNav (An et al., 2024) [TPAMI24]	11.78	3.95	72	66	59	11.99	4.71	65	57	49	12.87	5.12	63	55	48
	ETPNav*	11.35	3.93	72	66	59	11.40	4.69	64	57	49	12.72	5.10	63	55	48
	SC^2-WM	12.15	3.85	73	68	60	12.05	4.60	68	59	51	13.88	4.97	65	57	50
	HNR (Wang et al., 2024b) [CVPR24]	11.79	3.67	76	69	61	12.64	4.42	67	61	51	13.03	4.81	67	58	50
	HNR*	11.84	3.73	76	69	61	12.76	4.57	67	61	51	12.92	4.85	67	58	50
	SC^2-WM	12.09	3.28	80	71	64	12.89	4.25	70	66	54	13.42	4.90	71	62	53

Note: Following prior work, we report the results with different precision formats across camera configurations——integers for panoramic settings and two decimal places for monocular settings.

Once triggered, we update the world model using the self-supervised reconstruction objective $\mathcal{L}_{\text{rec}}$ (Eq.(13)):

$$\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} - \eta \cdot \nabla_{\boldsymbol{\Theta}} \mathcal{L}_{\text{rec}}, \tag{14}$$

where η denotes the adaptation learning rate and we denote all components of the world model without policies collectively by parameters $\boldsymbol{\Theta}$

This online update allows SC^2-WM to refine its internal dynamics in testing scenarios, working synergistically with plan refinement to enable effective navigation.

4. Experiments

To validate the effectiveness of our proposed method, we conduct experiments on two continuous environment benchmarks: R2R-CE and RxR-CE (Krantz et al., 2020), which are continuous reconstructions of the discrete R2R (Anderson et al., 2018b) and RxR (Ku et al., 2020) datasets. Please refer to the Appendix for additional details.

4.1. Experimental Setup

Datasets. The R2R-CE dataset (Krantz et al., 2020; Anderson et al., 2018b) consists of 5,611 shortest-path trajectories distributed across training, validation, and test sets. Each path is associated with an average of three English instructions. The trajectories feature a mean length of 9.89 meters, while the instructions have an average length of 32 words.The RxR-CE dataset (Krantz et al., 2020; Ku et al., 2020), which shares similar scene splits, presents a more challenging setting with a larger scale and multilingual instructions (English, Hindi, and Telugu). The instructions here are significantly longer, averaging 120 words.A key distinction lies in the agent dynamics: R2R-CE agents possess a chassis radius of 0.10 meters and are permitted to slide along obstacles. In contrast, RxR-CE agents have a larger radius of 0.18 meters and are strictly prohibited from sliding, making them more susceptible to collisions.

Evaluation Metrics. We employ a comprehensive set of metrics used in prior works(Anderson et al., 2018a;b; Ilharco et al., 2019): Trajectory Length (TL), Navigation Error (NE), Success Rate given Oracle stop policy (OSR), Success Rate (SR), Success weighted by Path Length (SPL), Normalized Dynamic Time Warping (NDTW), and Success weighted by NDTW (SDTW).

Implementation Details. We evaluate our method in both monocular and panoramic settings. For monocular navigation, we adopt the VLN-3DFF framework (Wang et al., 2025c), which leverages pretrained 3D Feature Fields (Wang et al., 2024b) to enable monocular agents to operate in environments originally designed for panoramic observations. This setting reflects practical deployment constraints, where monocular cameras offer advantages in cost and energy efficiency. For panoramic navigation, we follow the standard VLN-CE protocol (Krantz et al., 2020; 2021), with

Table 2. Experimental results on RxR-CE dataset.

Camera	Methods	Val Unseen				
		NE ↓	SR	SPL	NDTW	SDTW
Monocular	LAW (Raychaudhuri et al., 2021)	10.87	8.0	8.0	-	-
	CM^2 (Georgakis et al., 2022)	8.98	14.4	9.2	-	-
	WS-MGMap (Chen et al., 2022a)	9.83	15.0	12.1	-	-
	NaVid (Zhang et al., 2024)	8.41	23.8	32.2	-	-
	A^2-Nav (Chen et al., 2023)	-	16.8	6.3	-	-
	NavMorph (Yao et al., 2025)	8.85	30.8	22.8	44.2	23.3
	VLN-3DFF (Wang et al., 2025c)	8.79	25.5	18.1	-	-
	VLN-3DFF*	9.4	26.7	20.1	42.9	20.4
	SC^2-WM	8.36	35.8	27.2	44.9	26.5
Panoramic	LAW-Pano (Raychaudhuri et al., 2021)	11.04	10	9	-	-
	Seq2Seq (Anderson et al., 2018b)	12.10	14	12	31	11
	CWP-CMA (Hong et al., 2022)	8.76	27	22	47	-
	CWP-BERT (Hong et al., 2022)	8.98	27	23	47	-
	AO-Planner (Chen et al., 2024)	7.06	43	30	50	-
	Reborn (An et al., 2022b)	5.98	49	42	63	42
	NavMorph (Yao et al., 2025)	5.70	58	49	65	49
	ETPNav (An et al., 2024)	5.64	55	45	62	45
	ETPNav*	5.96	55	45	61	45
	SC^2-WM	5.57	57	46	64	47
	HNR (Wang et al., 2024b)	5.51	56	47	64	47
	HNR*	5.75	56	47	63	47
	SC^2-WM	5.72	60	50	67	50

Note: * indicates experimental results that we have reproduced in this work.

12 RGB-D images captured at 30° intervals per location. We integrate our method with ETPNav (An et al., 2024) and HNR (Wang et al., 2024b) to verify its generalizability across different architectures. All experiments follow the online VLN protocol (Gao et al., 2024) with batch size 1 during evaluation. Our model is implemented in PyTorch and trained on a single NVIDIA RTX 3090 GPU. Additional details are provided in the Appendix.

4.2. Comparison with State-of-the-art VLN Models

R2R-CE. Table 1 presents a comprehensive comparison between our proposed SC^2-WM and previous state-of-the-art methods on R2R-CE datasets. Our method significantly outperforms the strong baseline, VLN-3DFF, across multiple metrics. On the Val Unseen split, we achieve remarkable gains of 7.1% in Success Rate (SR) and nearly 7.8% in Sucess Path Length (SPL). Simultaneously, the method demonstrates a sharp reduction in Trajectory Length (TL), with an average decrease of 5 meters. These gains generalize robustly to the Test Unseen split, where SC^2-WM surpasses baseline by 3.2% in SR and 3.5% in SPL. Compared with the recent method g3D-LF, our approach achieves superior or comparable performance on most metrics. It is worth noting that g3D-LF leverages additional 3D representations, whereas our method does not. Besides, recent monoVLN method (Lu et al., 2025) employs more powerful 3DGS-based feature fields and achieves superior performance. Since monoVLN has not been open-sourced, we are unable to adopt it as a base model for world model construction. Incorporating more expressive 3D representations will be explored as future work.

We attribute the reduction in trajectory length directly to our correction mechanism. By anticipating potential future observations before executing an action, the agent effectively prunes useless exploration and avoids suboptimal steps. Regarding the improvements in SR and SPL, these stem from the intrinsic nature of the predictive module, alongside the effective utilization of historical memory. The objective of successfully predicting future visual features implicitly compels the model to develop a deeper cognitive understanding of the environment; this enhanced internal representation indirectly strengthens the agent's overall decision-making capability. Furthermore, our online Test-Time Adaptation (TTA) plays a critical role in generalization, as it continuously refines the World Model's predictive accuracy to maintain high environmental awareness even in unfamiliar, unseen settings. In the panoramic setting, SC^2-WM achieves the best overall results on the Val Unseen split, outperforming HNR by +5% SR and +3% SPL, indicating more stable and goal-aligned decision making.

Table 3. Ablation Study of the proposed World Model. Best results are highlighted in **bold**.

Model	State-C	VCM	Model-C	R2R-CE Val Unseen						
				TL↓	NE↓	OSR	SR	SPL	NDTW	SDTW
Base model	-	-	-	26.16	6.05	54.92	43.77	29.39	40.94	29.30
SC^2-WM	✓			21.75	5.69	56.22	48.34	33.02	45.36	32.82
	✓	✓		18.73	5.46	**59.22**	49.86	36.05	48.10	34.62
	✓	✓	✓†	18.43	5.43	58.67	50.30	36.58	48.66	35.37
	✓	✓	✓	**17.65**	**5.37**	58.84	**50.90**	**37.17**	**49.31**	**35.70**
SC^2-WM *w/o* $\mathcal{L}_{\text{rec}}$	✓	✓	✓	18.12	5.55	57.37	48.89	34.58	47.62	33.91

Note: State-C = State-Level Correction, Model-C = Model-Level Correction. † Fixed-interval adaptation with $T = 2$ instead of feedback-triggered adaptation.

RxR-CE. Table 2 extends our evaluation to the RxR-CE dataset, where SC^2-WM achieves comprehensive improvements over VLN-3DFF. We observe substantial gains of 9.1% in SR and 7.1% in SPL, alongside a consistent reduction in average trajectory length (8.36 m vs 9.41 m). Considering the strict sequentiality and temporal complexity of RxR instructions, these results underscore the necessity of our correction mechanism, which ensures the agent adheres closer to the described path, yielding higher path fidelity as evidenced by the significant boost in SDTW (26.5% vs 20.4%). As for panoramic setting, when built upon ETPNav, our method improves SR from 55% to 57% and SPL from 45% to 47% on the val-unseen split, while reducing TL from 5.96 to 5.57, indicating more efficient navigation paths. Notably, the consistent improvements across both frameworks demonstrate that our self-correcting mechanism is complementary to existing approaches and effectively mitigates error accumulation through closed-loop feedback.

4.3. Further Remarks

Ablation Study of the Proposed World Model. To investigate the efficacy of each component in SC^2-WM, we conduct ablation studies by incrementally incorporating the feedback-guided plan refinement ('State-C'), the visual calibration module ('VCM'), and the conditional world-aware adaptation ('Model-C'). Here, state-level correction refers to feedback-guided plan refinement that adjusts latent repre-

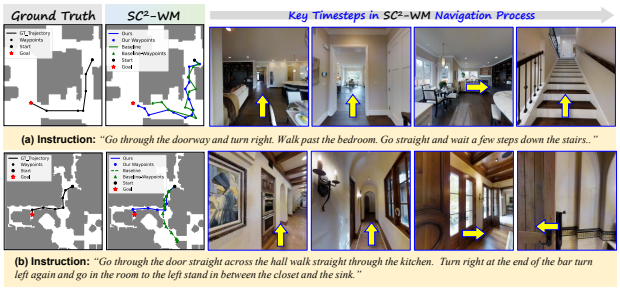

Figure 3. **Qualitative results on R2R-CE unseen set.** Comparison between our SC^2-WM (blue) and baseline VLN-3DFF (green) against ground truth (black). Yellow arrows indicate the agent's moving direction.

Figure 4. **Real-world deployment on a Unitree GO2 quadruped robot.** SC^2-WM enables robust vision-and-language navigation in indoor environments, leveraging internal feedback for self-correction under complex instructions.

sentations before action execution, while model-level correction refers to conditional world-aware adaptation that selectively updates the world model at test time. Table 3 details the contribution of each component on the R2R-CE Val Unseen split. **(1) Effect of State-Level Self-Correction:** Introducing feedback-guided plan refinement leads to a direct increase in SR from 43.77% to 48.34%, alongside a reduction in TL (from 26.16 m to 21.75 m). This indicates that the state-level correction effectively identifies and prunes erroneous steps by adjusting latent states based on world-model foresight. **(2) Impact of VCM:** The integration of the VCM module triggers a substantial performance leap, reducing TL to 18.73 m and boosting SPL to 36.05%. Episodic Memory in VCM provides the world model with essential historical context, enriching its foresight capability and enabling more accurate feedback signals for plan refinement. **(3) Benefit of Model-Level Correction:** Conditional world-aware adaptation further elevates performance by selectively updating the world model online. By adapting the predictive module to unseen environments, model-level correction achieves the best overall performance with an SR of 50.9% and the lowest TL of 17.65 m, validating the importance of dynamic adaptation for generalization. Notably, compared to the fixed-interval update mechanism adopted in (Gao et al., 2024), where we set $T = 2$ to match the average update frequency of our method for fair comparison, our feedback-triggered approach yields higher SR and SPL, demonstrating that selective adaptation based on internal feedback signals outperforms naive periodic updates. Additionally, removing the reconstruction loss $\mathcal{L}_{\text{rec}}$ (Eq.(13)) leads to performance degradation, confirming that $\mathcal{L}_{\text{rec}}$ helps learn effective latent representations for accurate world model predictions.

Qualitative Analysis. Figure 3 presents qualitative comparisons on the VLN-CE task using the R2R-CE dataset. We visualize two navigation episodes with varying instruction complexity. In both cases, our SC^2-WM (blue) produces trajectories that closely align with the ground truth (black), while the baseline method (green) tends to deviate from the intended path after a few steps. Notably, in Figure 3(b) where the instruction involves multiple turns and spatial references, our method maintains correct navigation throughout the episode, whereas the baseline fails to recover after early mistakes. These results suggest that our dual world-model design enables better spatial reasoning and more robust long-horizon navigation.

Real-world Experiments. We deploy SC^2-WM on a physical robotic platform to validate its effectiveness beyond simulation. The platform consists of a Unitree GO2 quadruped robot equipped with an Intel RealSense D435i RGB-D camera for visual perception. Given natural language navigation instructions, the robot is required to navigate through indoor environment. As shown in Figure 4, SC^2-WM demonstrates robust navigation behavior across varying conditions, highlighting the benefit of closed-loop self-correction for real-world deployment. Details are provided in the Appendix.

5. Conclusion

We presented SC^2-WM, a self-correcting world model framework that enables closed-loop decision making for vision-and-language navigation. By deriving internal feedback from world-model foresight, our approach performs state-level plan refinement and model-level adaptation, providing a principled way to mitigate error accumulation under partial observability. The current framework opens several promising directions for future research. The quality of internal feedback is tied to world-model expressiveness, suggesting potential benefits from incorporating uncertainty-aware or structured world models. Additionally, the test-time adaptation mechanism could be further optimized for efficiency to enable deployment on resource-constrained platforms. We also plan to extend the framework to longer-horizon planning and multi-agent scenarios, and explore tighter integration between language grounding and world modeling for richer semantic feedback.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

An, D., Qi, Y., Li, Y., Huang, Y., Wang, L., Tan, T., and Shao, J. Bevbert: Topo-metric map pre-training for language-guided navigation. *arXiv preprint arXiv:2212.04385*, 2022a.

An, D., Wang, Z., Li, Y., Wang, Y., Hong, Y., Huang, Y., Wang, L., and Shao, J. 1st place solutions for rxr-habitat vision-and-language navigation competition. *arXiv preprint arXiv:2206.11610*, 2022b.

An, D., Wang, H., Wang, W., Wang, Z., Huang, Y., He, K., and Wang, L. Etpnav: Evolving topological planning for vision-language navigation in continuous environments. *IEEE TPAMI*, 2024.

Anderson, P., Chang, A., Chaplot, D. S., Dosovitskiy, A., Gupta, S., Koltun, V., Kosecka, J., Malik, J., Mottaghi, R., Savva, M., et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018a.

Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., and Van Den Hengel, A. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, pp. 3674–3683, 2018b.

Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., and Sivic, J. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, pp. 5297–5307, 2016.

Banerjee, S., Thomason, J., and Corso, J. The robotslang benchmark: Dialog-guided robot localization and navigation. In *CoRL*, pp. 1384–1393, 2021.

Bu, Q., Zeng, J., Chen, L., Yang, Y., Zhou, G., Yan, J., Luo, P., Cui, H., Ma, Y., and Li, H. Closed-loop visuomotor control with generative expectation for robotic manipulation. *NeurIPS*, 37:139002–139029, 2024.

Chen, J., Lin, B., Liu, X., Liang, X., and Wong, K.-Y. K. Affordances-oriented planning using foundation models for continuous vision-language navigation. *arXiv preprint arXiv:2407.05890*, 2024.

Chen, P., Ji, D., Lin, K.-L. C., Zeng, R., Li, T. H., Tan, M., and Gan, C. Weakly-supervised multi-granularity map learning for vision-and-language navigation. In *NeurIPS*, pp. 38149–38161, 2022a.

Chen, P., Sun, X., Zhi, H., Zeng, R., Li, T. H., Liu, G., Tan, M., and Gan, C. A^2 nav: Action-aware zero-shot robot navigation by exploiting vision-and-language ability of foundation models. *arXiv preprint arXiv:2308.07997*, 2023.

Chen, S., Guhur, P.-L., Tapaswi, M., Schmid, C., and Laptev, I. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *CVPR*, 2022b.

Chen, S., Guhur, P.-L., Tapaswi, M., Schmid, C., and Laptev, I. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *CVPR*, pp. 16537–16547, 2022c.

Cheng, A.-C., Ji, Y., Yang, Z., Gongye, Z., Zou, X., Kautz, J., Bıyık, E., Yin, H., Liu, S., and Wang, X. Navila: Legged robot vision-language-action model for navigation. *arXiv preprint arXiv:2412.04453*, 2024.

Chi, X., Zhang, H., Fan, C.-K., Qi, X., Zhang, R., Chen, A., min Chan, C., Xue, W., Luo, W., Zhang, S., and Guo, Y. Eva: An embodied world model for future video anticipation, 2024. URL https://arxiv.org/abs/2410.15461.

Chi, X., Jia, P., Fan, C.-K., Ju, X., Mi, W., Zhang, K., Qin, Z., Tian, W., Ge, K., Li, H., et al. Wow: Towards a world omniscient world model through embodied interaction. *arXiv preprint arXiv:2509.22642*, 2025.

Dai, G., Zhao, J., Chen, Y., Qin, Y., Zhao, H., Xie, G., Yao, Y., Shu, X., and Li, X. Unitedvln: Generalizable gaussian splatting for continuous vision-language navigation. *arXiv preprint arXiv:2411.16053*, 2024.

Dey, A. and Bhasin, S. Adaptive output feedback mpc with guaranteed stability and robustness. *IEEE Transactions on Automatic Control*, 2025.

Dosovitskiy, A. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Gao, J., Yao, X., and Xu, C. Fast-slow test-time adaptation for online vision-and-language navigation. In *ICML*, pp. 14902–14919, 2024.

Gao, J., Yao, X., Rui, Y., and Xu, C. Building embodied evoagent: A brain-inspired paradigm for bridging multimodal large models and world models. In *ACM MM*, pp. 3280–3289, 2025.

Georgakis, G., Schmeckpeper, K., Wanchoo, K., Dan, S., Miltsakaki, E., Roth, D., and Daniilidis, K. Cross-modal map learning for vision and language navigation. In *CVPR*, pp. 15439–15449, 2022.

Ha, D. and Schmidhuber, J. World models. *arXiv preprint arXiv:1803.10122*, 2(3), 2018.

Hafner, D., Pasukonis, J., Ba, J., and Lillicrap, T. Mastering diverse control tasks through world models. *Nature*, pp. 1–7, 2025.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Hong, Y., Wang, Z., Wu, Q., and Gould, S. Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation. In *CVPR*, pp. 15418–15428, 2022.

Huang, C., Tang, L., Zhan, Z., Yu, L., Zeng, R., Liu, Z., Wang, Z., and Li, J. UNeMo: Collaborative visual-language reasoning and navigation via a multimodal world model. *arXiv preprint arXiv:2511.18845*, 2025.

Huang, Y. and Rao, R. P. Predictive coding. *Wiley Interdisciplinary Reviews: Cognitive Science*, 2(5):580–593, 2011.

Hutchinson, S., Hager, G. D., and Corke, P. I. A tutorial on visual servo control. *IEEE transactions on robotics and automation*, 12(5):651–670, 2002.

Ilharco, G., Jain, V., Ku, A., Ie, E., and Baldridge, J. General evaluation for instruction conditioned navigation using dynamic time warping. *arXiv preprint arXiv:1907.05446*, 2019.

Krantz, J. and Lee, S. Sim-2-sim transfer for vision-and-language navigation in continuous environments. In *ECCV*, pp. 588–603, 2022.

Krantz, J., Wijmans, E., Majumdar, A., Batra, D., and Lee, S. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *ECCV*, pp. 104–120, 2020.

Krantz, J., Gokaslan, A., Batra, D., Lee, S., and Maksymets, O. Waypoint models for instruction-guided navigation in continuous environments. In *ICCV*, pp. 15162–15171, 2021.

Ku, A., Anderson, P., Patel, R., Ie, E., and Baldridge, J. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *EMNLP*, pp. 4392–4412, 2020.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

Li, C., Liu, J., Wang, G., Li, X., Chen, S., Heng, L., Xiong, C., Ge, J., Zhang, R., Zhou, K., et al. A self-correcting vision-language-action model for fast and slow system manipulation. *arXiv preprint arXiv:2405.17418*, 2024.

Lin, B., Nie, Y., Wei, Z., Chen, J., Ma, S., Han, J., Xu, H., Chang, X., and Liang, X. Navcot: Boosting llm-based vision-and-language navigation via learning disentangled reasoning. *IEEE TPAMI*, 2025.

Liu, Y. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 364, 2019.

Long, Y., Cai, W., Wang, H., Zhan, G., and Dong, H. Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. *arXiv preprint arXiv:2406.04882*, 2024.

Lu, R., Zhou, Y., Cheng, H., Meng, J., and Zheng, W.-S. monovln: Bridging the observation gap between monocular and panoramic vision and language navigation. In *ICCV*, pp. 9477–9486, October 2025.

Min, C., Zhao, D., Xiao, L., Zhao, J., Xu, X., Zhu, Z., Jin, L., Li, J., Guo, Y., Xing, J., et al. Driveworld: 4d pre-trained scene understanding via world models for autonomous driving. In *CVPR*, pp. 15522–15533, 2024.

Qi, Y., Wu, Q., Anderson, P., Wang, X., Wang, W. Y., Shen, C., and Hengel, A. v. d. Reverie: Remote embodied visual referring expression in real indoor environments. In *CVPR*, pp. 9982–9991, 2020.

Qiao, Y., Qi, Y., Hong, Y., Yu, Z., Wang, P., and Wu, Q. HOP+: History-Enhanced and Order-Aware Pre-Training for Vision-and-Language Navigation. *IEEE TPAMI*, 45 (7):8524–8537, 2023.

Raychaudhuri, S., Wani, S., Patel, S., Jain, U., and Chang, A. X. Language-aligned waypoint (law) supervision for vision-and-language navigation in continuous environments. In *EMNLP*, pp. 4018–4028, 2021.

Ren, X., Lu, Y., Cao, T., Gao, R., Huang, S., Sabour, A., Shen, T., Pfaff, T., Wu, J. Z., Chen, R., Kim, S. W., Gao, J., Leal-Taixe, L., Chen, M., Fidler, S., and Ling, H. Cosmos-drive-dreams: Scalable synthetic driving data generation with world foundation models, 2025. URL https://arxiv.org/abs/2506.09042.

Russell, L., Hu, A., Bertoni, L., Fedoseev, G., Shotton, J., Arani, E., and Corrado, G. Gaia-2: A controllable multi-view generative world model for autonomous driving. *arXiv preprint arXiv:2503.20523*, 2025.

Sendai, K., Alvarez, M., Matsushima, T., Matsuo, Y., and Iwasawa, Y. Leave no observation behind: Real-time correction for vla action chunks, 2025. URL https://arxiv.org/abs/2509.23224.

Shi, X., Li, Z., Lyu, W., Xia, J., Dayoub, F., Qiao, Y., and Wu, Q. Smartway: Enhanced waypoint prediction and backtracking for zero-shot vision-and-language navigation. *arXiv preprint arXiv:2503.10069*, 2025.

Song, X., Chen, W., Liu, Y., Chen, W., Li, G., and Lin, L. Towards long-horizon vision-language navigation: Platform, benchmark and method. In *CVPR*, 2025.

Tan, H. and Bansal, M. Lxmert: Learning cross-modality encoder representations from transformers. *arXiv preprint arXiv:1908.07490*, 2019.

Valevski, D., Leviathan, Y., Arar, M., and Fruchter, S. Diffusion models are real-time game engines, 2024. URL https://arxiv.org/abs/2408.14837.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

Wang, H., Liang, W., Gool, L. V., and Wang, W. Dreamwalker: Mental planning for continuous vision-language navigation. In *ICCV*, pp. 10873–10883, 2023a.

Wang, H., Liang, W., Van Gool, L., and Wang, W. Dreamwalker: Mental planning for continuous vision-language navigation. In *ICCV*, pp. 10873–10883, 2023b.

Wang, S., Wang, Y., Li, W., Cai, X., Wang, Y., Chen, M., Wang, K., Su, Z., Li, D., and Fan, Z. Aux-think: Exploring reasoning strategies for data-efficient vision-language navigation. *arXiv preprint arXiv:2505.11886*, 2025a.

Wang, Y., Sun, Z., Zhang, J., Xian, Z., Biyik, E., Held, D., and Erickson, Z. Rl-vlm-f: Reinforcement learning from vision language foundation model feedback. *arXiv preprint arXiv:2402.03681*, 2024a.

Wang, Z. and Lee, G. H. g3d-lf: Generalizable 3d-language feature fields for embodied tasks. In *CVPR*, pp. 14191–14202, June 2025.

Wang, Z., Li, X., Yang, J., Liu, Y., and Jiang, S. Gridmm: Grid memory map for vision-and-language navigation. In *ICCV*, pp. 15625–15636, 2023c.

Wang, Z., Li, X., Yang, J., Liu, Y., Hu, J., Jiang, M., and Jiang, S. Lookahead exploration with neural radiance representation for continuous vision-language navigation. In *CVPR*, pp. 13753–13762, 2024b.

Wang, Z., Lee, S., and Lee, G. H. Dynam3d: Dynamic layered 3d tokens empower vlm for vision-and-language navigation. In *NeurIPS*, 2025b.

Wang, Z., Li, X., Yang, J., and Jiang, S. Sim-to-real transfer via 3d feature fields for vision-and-language navigation. In *CoRL*, 2025c.

Wei, M., Wan, C., Peng, J., Yu, X., Yang, Y., Feng, D., Cai, W., Zhu, C., Wang, T., Pang, J., et al. Ground slow, move fast: A dual-system foundation model for generalizable vision-and-language navigation. *arXiv preprint arXiv:2512.08186*, 2025a.

Wei, M., Wan, C., Yu, X., Wang, T., Yang, Y., Mao, X., Zhu, C., Cai, W., Wang, H., Chen, Y., et al. Streamvln: Streaming vision-and-language navigation via slowfast context modeling. *arXiv preprint arXiv:2507.05240*, 2025b.

Yao, X., Gao, J., and Xu, C. Navmorph: A self-evolving world model for vision-and-language navigation in continuous environments. In *ICCV*, pp. 5536–5546, 2025.

Yuan, S., Song, K., Chen, J., Tan, X., Li, D., and Yang, D. Evoagent: Towards automatic multi-agent generation via evolutionary algorithms. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6192–6217, 2025.

Zhang, J., Wang, K., Xu, R., Zhou, G., Hong, Y., Fang, X., Wu, Q., Zhang, Z., and He, W. Navid: Video-based vlm plans the next step for vision-and-language navigation. In *RSS*, 2024.

Zhang, Z., Zhu, W., Pan, H., Wang, X., Xu, R., Sun, X., and Zheng, F. Activevln: Towards active exploration via multi-turn rl in vision-and-language navigation. *arXiv preprint arXiv:2509.12618*, 2025.

§ A presents the guidelines for our source code. § B presents more details about evaluation metrics. Implementation details for experiments are provided in § C, and further comparisons against state-of-the-art methods are shown in § D. Finally, we present real-world verification to validate the effectiveness of our method in § E.

A. Source Code

The source code to reproduce our experimental results is included in the supplementary material (6618-supp.zip) under the `./code` directory. For more details, please refer to the README file therein.

B. Evaluation Metrics for VLN-CE agents

We follow previous approaches (Anderson et al., 2018b;a; Ilharco et al., 2019) and adopt the standard metrics for evaluating VLN-CE agents:

- **TL (Trajectory length)** measures the average length of the predicted navigation trajectories.
- **NE (Navigation Error)** measures the average distance (in meter) between the agent's final position in the predicted trajectory and the target in the ground truth.
- **SR (Success Rate)** is the proportion of the agent stopping in the predicted route within a threshold distance (set as 3 meters) of the goal in the reference route.
- **OSR (Oracle Success Rate)** is the proportion of the closest point in the predicted trajectory to the target in the reference trajectory within a threshold distance.
- **SPL (Success weighted by Path Length)** is a comprehensive metric method integrating SR and TL that takes both effectiveness and efficiency into account.
- **NDTW (Normalized Dynamic Time Warping)** measures the normalized cumulative distance between reference path and agent position.
- **SDTW (Success weighted by normalized Dynamic Time Warping)** is a comprehensive metric method integrating NDTW and SR that takes both path efficiency and task completion into account.

C. Implementation Details

The Baseline Framework. In conventional panoramic VLN-CE frameworks (An et al., 2024; Wang et al., 2023c), the agent perceives its surroundings through multi-view RGB-D panoramas captured at 30-degree intervals at each timestep t. These panoramic observations are processed by a trained waypoint prediction module(Hong et al., 2022) to identify navigable waypoints. The VLN model then encodes both the visual features of these waypoints and their spatial information (relative direction and distance) to construct a topological map. This map is subsequently integrated with the navigation instruction via the Cross-Modal Graph Transformer(Chen et al., 2022c; An et al., 2024), which selects the optimal waypoint as the agent's next navigation goal.

Monocular VLN-CE settings rely on a single RGB-D camera, which presents challenges in waypoint estimation due to the lack of full panoramic coverage. To address this, an enhanced waypoint predictor (Wang et al., 2025c) utilizes a semantic traversability map and 3D feature fields to infer viable waypoints, ensuring effective decision-making even with limited field-of-view.

Model Configuration. Following the previous baseline model (An et al., 2024; Wang et al., 2025c), we utilize CLIP-pretrained ViT-B/32 (Dosovitskiy, 2020) for RGB feature extraction, while depth information is processed through a point-goal navigation pretrained ResNet-50 (He et al., 2016). The framework maintains encoder depths of 2, 9, and 4 layers for panoramic, textual, and cross-modal graph components respectively, aligned with (Hong et al., 2022; Georgakis et al., 2022). Other hyperparameters are the same as LXMERT (Tan & Bansal, 2019) on the R2R-CE dataset and pre-trained RoBerta (Liu, 2019) for the multilingual RxR-CE dataset. The camera's HFOV is set to 90° for R2R-CE and 79° for RxR-CE.

Experimental Details. We conduct our experiments using distinct initialization and training strategies for the R2R-CE and RxR-CE datasets. For R2R-CE, we initialize the model with weights pretrained on the R2R-CE dataset for 25,000 iterations. We employ a hierarchical training strategy consisting of two stages: First, we jointly train all modules with a learning rate of 1×10^{-5} for 12,000 episodes. Subsequently, we freeze the language encoder and fine-tune the remaining components with a reduced learning rate of 4×10^{-6} for an additional 10,000 episodes. For RxR-CE, we initialize our framework using the pretrained checkpoint from VLN-3DFF (Wang et al., 2025c). The model is then trained with a learning rate of 1×10^{-5} for 14,000 episodes.

Temporal Prior in Memory of VCM module. To explicitly incorporate temporal locality and prioritize recent navigational contexts, we inject a learnable temporal bias into the cross-attention mechanism of the memory $\mathcal{M}_t$. We introduce a learnable scalar parameter λ (initialized to 0.1) to regulate the attention distribution based on the temporal distance between the current time step t and a historical memory step k. A bias term, formulated as $B_{t,k} = -|\lambda| \cdot (t-k)$, is added to the raw attention logits prior to the softmax normalization. This linear decay effectively penalizes older historical observations, encouraging the agent to assign higher attention weights to the most immediate predecessor steps during the reasoning process.

Spatio-temporal Encodings in Memory of VCM Module. To effectively ground historical observations within the agent's current frame of reference, we enhance the visual features of memory items with explicit spatial and temporal encodings prior to the attention mechanism:

(i) Relative Spatial Encoding. We compute a 7-dimensional relative geometric feature vector $\mathbf{g}_{t,k} \in \mathbb{R}^7$ for each historical node k with respect to the agent's current pose at step t. This vector consists of the relative heading and elevation (represented by sine and cosine values) and the normalized Euclidean distance. These geometric features are projected into the model's hidden dimension D via a linear layer followed by Layer Normalization:

$$\mathbf{e}_{t,k}^{(pos)} = \text{LayerNorm}(\mathbf{W}_p \mathbf{g}_{t,k} + \mathbf{b}_p), \tag{15}$$

where $\mathbf{W}_p$ and $\mathbf{b}_p$ are learnable parameters.

(ii) Step (Temporal) Embedding. To encode the sequential order and temporal distance of visited nodes, we introduce a discrete step embedding. Let $\delta t = \min(t-k, T_{max})$ denote the clamped time difference between the current step t and the memory step k. We utilize a learnable lookup table $\mathbf{E}_{step}$ to retrieve a dense vector representation for this time lag:

$$\mathbf{e}_{t,k}^{(step)} = \mathbf{E}_{step}[\delta t]. \tag{16}$$

Both $\mathbf{e}_{t,k}^{(pos)}$ and $\mathbf{e}_{t,k}^{(step)}$ are added element-wise to the visual features of the corresponding memory item, thereby fusing spatiotemporal context into the visual representation.

D. Complementary Experiments

D.1. Full Results

In our main paper, we provide representative comparison results on the R2R-CE (Krantz et al., 2020) and RxR-CE (Krantz et al., 2020) benchmarks due to space constraints. Here, we present the complete results across the 'validation seen', 'validation unseen', and 'test unseen' splits of these benchmarks, including comparisons with a broader range of state-of-the-art methods, as detailed in Table 4 and Table 5. Our proposed SC^2-WM method enhances its ability to perform navigation decision based on self-correcting with closed-loop feedback, effectively handling complex navigation tasks even with monocular input. Note that the recent monoVLN method (Lu et al., 2025) employs more powerful 3DGS-based feature fields and achieves superior performance. Since monoVLN has not been open-sourced, we are unable to adopt it as a base model for world model construction. Incorporating more expressive 3D representations will be explored as future work. Besides, compared with monoVLN, our method achieves better performance on the RxR dataset.

D.2. Discussion on Model Paradigm

While we include MLLM-based VLN methods in the full results (Table 4), our work focuses on a lightweight world-model-based framework rather than incorporating large-scale multimodal language models. Although recent MLLM-based approaches demonstrate strong semantic reasoning, they typically incur substantial computational overhead and

remain limited in online adaptability during execution. In contrast, prior studies have shown that improving internal state representations and predictive dynamics can effectively enhance decision stability under partial observability (Yao et al., 2025; Wang & Lee, 2025; Huang et al., 2025; An et al., 2024). Building on this line of work, our design introduces internal feedback and self-correction within a compact world model, enabling robust execution-time regulation without relying on large-scale models. It is worth noting that the model proposed in this paper requires only a single NVIDIA RTX 3090 GPU and less than 40 hours of training to achieve strong performance, whereas such computational resources are typically insufficient to meet the demands of methods based on large-scale multimodal models.

Table 4. Experimental results on R2R-CE dataset. Results better than **base model** are shown in blue. Best results for **both panoramic and monocular settings** are underlined. * indicates experimental results that we have reproduced. † Methods based on large language/vision-language models.

Camera	Methods	Val Seen					Val Unseen					Test Unseen				
		TL ↓	NE ↓	OSR	SR	SPL	TL ↓	NE ↓	OSR	SR	SPL	TL ↓	NE ↓	OSR	SR	SPL
Monocular	LAW (Raychaudhuri et al., 2021) [EMNLP21]	9.34	6.35	49	40	37	8.89	6.83	44	35	31	9.67	7.69	28	38	25
	CM² (Georgakis et al., 2022) [CVPR22]	12.05	6.10	50.7	42.9	34.8	11.54	7.02	41.5	34.3	27.6	13.90	7.70	39	31	24
	WS-MGMap (Chen et al., 2022a) [NeurIPS22]	10.12	5.65	51.7	46.9	43.4	10.00	6.28	47.6	38.9	34.3	12.30	7.11	45	35	28
	NaVid (Zhang et al., 2024) [RSS24]	-	-	-	-	-	-	5.47	49.1	37.4	35.9	-	-	-	-	-
	ETPNav/p (Wang et al., 2025c) [CoRL24]	-	-	-	-	-	-	6.81	42.4	32.9	23.1	-	-	-	-	-
	NavMorph (Yao et al., 2025) [ICCV2025]	20.03	4.58	62.7	55.8	38.9	22.54	5.75	56.9	47.9	33.2	24.75	6.01	54.5	45.7	30.2
	g3D-LF (Wang & Lee, 2025) [CVPR2025]	-	-	-	-	-	-	5.70	59.5	47.2	34.6	-	6.00	57.5	46.3	32.2
	monoVLN (Lu et al., 2025) [CVPR2025]	-	-	-	-	-	-	4.61	62.4	54.8	44.4	-	4.97	60.5	53.6	44.9
	NaVILA† (Cheng et al., 2024) [arxiv2025]	-	-	-	-	-	-	5.43	62.5	54.0	49.0	-	-	-	-	-
	UniNaVid† (Zhang et al., 2024) [arxiv2025]	-	-	-	-	-	-	5.58	53.3	47.0	42.7	-	-	-	-	-
	StreamVLN† (Wei et al., 2025b) [arxiv2025]	-	-	-	-	-	-	4.98	64.2	56.9	51.9	-	-	-	-	-
	DualVLN† (Wei et al., 2025a)[arxiv2025]	-	-	-	-	-	-	4.05	70.7	64.3	58.5	-	-	-	-	-
	VLN-3DFF (Wang et al., 2025c) [CoRL24]	-	-	-	-	-	-	5.95	55.8	44.9	30.4	-	6.24	54.4	43.7	28.9
	VLN-3DFF*	22.90	4.92	62.1	52.7	36.7	26.16	6.05	54.9	43.8	29.4	26.02	6.22	54.7	43.8	28.6
	SC²-WM	17.26	4.53	64.3	56.0	41.9	17.65	5.37	58.8	50.9	37.2	21.68	6.04	57.1	47.0	32.1
Panoramic	Seq2Seq (Anderson et al., 2018b) [CVPR18]	9.26	7.12	46	37	35	8.64	7.37	40	32	30	8.85	7.91	36	28	25
	Sim2Sim (Krantz & Lee, 2022) [ECCV22]	11.18	4.67	61	52	44	10.69	6.07	52	43	36	11.43	6.17	52	44	37
	CWP-CMA (Hong et al., 2022) [CVPR22]	11.47	5.20	61	51	45	10.90	6.20	52	41	36	11.85	6.30	49	38	33
	CWP-BERT (Hong et al., 2022)[CVPR22]	12.50	5.02	59	50	44	12.23	5.74	53	44	39	13.51	5.89	51	42	36
	DREAMW (Wang et al., 2023a) [ICCV23]	11.60	4.09	59	66	48	11.30	5.53	49	59	44	11.80	5.48	49	57	44
	GridMM (Wang et al., 2023c) [ICCV23]	12.69	4.21	69	59	51	13.36	5.11	61	49	41	13.31	5.64	56	46	39
	BEVBert (An et al., 2022a) [ICCV23]	13.98	3.77	73	68	60	13.27	4.57	67	59	50	15.31	4.70	67	59	50
	InstructNav† (Long et al., 2024) [arxiv2024]	-	-	-	-	-	7.74	6.89	-	31	24	-	-	-	-	-
	FSTTA (Gao et al., 2024) [ICML24]	12.39	4.25	69	58	50	11.58	5.27	58	48	42	13.17	5.84	55	46	38
	SmartWay† (Shi et al., 2025) [arxiv2025]	-	-	-	-	-	13.09	7.01	51	29	22	-	-	-	-	-
	Aux-Think† (Wang et al., 2025a) [arxiv2025]	-	-	-	-	-	-	6.01	52	46	41	-	-	-	-	-
	Dynam3D† (Wang et al., 2025b) [ICCV2025]	-	-	-	-	-	-	5.34	62	53	46	-	5.53	60	51	45
	NavMorph (Yao et al., 2025) [ICCV2025]	11.76	3.66	78	70	62	12.68	4.37	68	64	53	12.69	4.69	68	60	52
	g3D-LF (Wang & Lee, 2025) [CVPR2025]	-	-	-	-	-	-	4.53	68	61	52	-	4.78	68	58	51
	ETPNav (An et al., 2024) [TPAMI24]	11.78	3.95	72	66	59	11.99	4.71	65	57	49	12.87	5.12	63	55	48
	ETPNav*	11.35	3.93	72	66	59	11.40	4.69	64	57	49	12.72	5.10	63	55	48
	SC²-WM	12.15	3.85	73	68	60	12.05	4.60	68	59	51	13.88	4.97	65	57	50
	HNR (Wang et al., 2024b) [CVPR24]	11.79	3.67	76	69	61	12.64	4.42	67	61	51	13.03	4.81	67	58	50
	HNR*	11.84	3.73	76	69	61	12.76	4.57	67	61	51	12.92	4.85	67	58	50
	SC²-WM	12.09	3.28	80	71	64	12.89	4.25	70	66	54	13.42	4.90	71	62	53

Note: Following prior work, we report the results with different precision formats across camera configurations——integers for panoramic settings and two decimal places for monocular settings.

Analysis of Optimization Objectives for Conditional World-Aware Adaptation. As detailed in Table 6, we investigate the effectiveness of three distinct self-supervised objectives for the adaptation of the world model during inference. We compare the Self-Entropy Loss, which minimizes the entropy of the scoring distribution to encourage high-confidence predictions; the Asynchronous Scoring Loss, which utilizes the posterior score from the subsequent step to supervise the current step under the assumption that later steps possess better context; and our proposed Foresight-Reconstruction Loss (FRL, Eq.(13)), which aligns the World Model's predicted next-step features with the actual observations. Experimental results indicate that the Self-Entropy Loss fails to improve performance, as it lacks grounded supervision and merely reinforces existing beliefs without correcting errors. Besides, the Asynchronous Scoring Loss achieves superior results compared to the w/o-adaptation baseline. However, the improvement is not significant due to the scoring function's lack of strict temporal consistency; influenced by extraneous factors like the absolute step index, the posterior score serves as a noisy supervision target. In contrast, FRL achieves superior performance by retaining objective consistency with the training phase. By leveraging the deviation between the model's 'imagination' and reality as a robust error signal, FRL ensures effective model updates and better generalization in unseen environments.

Ablation Study of Visual Memory Design. We further investigate the optimal size of the memory buffer, which stores

Table 5. Experimental results on RxR-CE datasets. Results better than the base model are shown in blue. Best results for the panoramic and monocular settings are each highlighted in bold.

Camera	Methods	Val Seen							Val Unseen							Test Unseen						
		TL ↓	NE ↓	OSR	SR	SPL	NDTW	SDTW	TL ↓	NE ↓	OSR	SR	SPL	NDTW	SDTW	TL ↓	NE ↓	OSR	SR	SPL	NDTW	SDTW
Monocular	LAW (Raychaudhuri et al., 2021)	**7.92**	11.94	20.0	7.0	6.0	-	-	**4.01**	10.87	21.0	8.0	8.0	-	-	-	-	-	-	-	-	-
	CM^2 (Georgakis et al., 2022)	-	-	-	-	-	-	-	12.29	8.98	25.3	14.4	9.2	-	-	-	-	-	-	-	-	-
	WS-MGMap (Chen et al., 2022a)	10.37	10.19	27.7	14.0	12.3	-	-	10.80	9.83	29.8	15.0	12.1	-	-	-	-	-	-	-	-	-
	NaVid (Zhang et al., 2024)	-	-	-	-	-	-	-	10.59	8.41	34.5	23.8	32.2	-	-	-	-	-	-	-	-	-
	A^2-Nav (Chen et al., 2023)	-	-	-	-	-	-	-	-	-	-	16.8	6.3	-	-	-	-	-	-	-	-	-
	NavMorph (Yao et al., 2025)	21.61	9.80	41.27	29.81	23.23	**44.51**	22.68	20.28	8.85	43.05	30.76	22.84	44.19	23.30	21.13	**9.81**	-	**24.93**	**16.82**	**33.71**	**15.64**
	monoVLN (Lu et al., 2025)	-	-	-	-	-	-	-	-	8.29	37.7	31.8	26.8	-	25.2	-	-	-	-	-	-	-
	VLN-3DFF (Wang et al., 2025c)	-	-	-	-	-	-	-	-	8.79	36.7	25.5	18.1	-	-	-	-	-	-	-	-	-
	VLN-3DFF*	18.91	9.87	40.54	27.72	20.61	42.37	20.94	16.21	9.41	38.40	26.66	20.11	42.91	20.36	**20.85**	10.19	-	23.41	15.43	32.38	14.75
	SC^2-WM	22.08	**9.39**	**44.83**	**31.66**	**23.97**	42.81	**23.65**	20.87	**8.36**	**48.38**	**35.77**	**27.20**	**44.93**	**26.47**	-	-	-	-	-	-	-
Panoramic	Seq2Seq (Anderson et al., 2018b)	-	-	-	-	-	-	-	7.33	12.1	-	13.93	11.96	30.86	11.01	-	12.10	-	13.93	11.96	30.86	11.01
	Reborn (An et al., 2022b)	-	5.69	-	52.43	45.46	66.27	44.47	-	5.98	-	48.60	42.05	63.35	41.82	-	7.10	-	45.82	38.82	55.43	38.42
	CWP-CMA (Hong et al., 2022)	-	-	-	-	-	-	-	-	8.76	-	26.59	22.16	47.05	-	**20.04**	10.4	-	24.08	19.07	37.39	18.65
	CWP-RecBERT (Hong et al., 2022)	-	-	-	-	-	-	-	-	8.98	-	27.08	22.65	46.71	-	20.09	10.4	-	24.85	19.61	37.30	19.05
	AO-Planner (Chen et al., 2024)	-	-	-	-	-	-	-	-	7.06	-	43.3	30.5	50.1	-	-	-	-	-	-	-	-
	LAW-Pano (Raychaudhuri et al., 2021)	**6.27**	12.07	17.0	9.0	9.0	-	-	**4.62**	11.04	16.0	10.0	9.0	-	-	-	-	-	-	-	-	-
	UnitedVLN (Dai et al., 2024)	-	**4.74**	-	**65.1**	52.9	69.4	53.6	-	**5.48**	-	57.9	45.9	63.9	48.1	-	-	-	-	-	-	-
	NavMorph (Yao et al., 2025)	20.80	5.10	**67.88**	64.95	**54.17**	**70.94**	**54.82**	21.33	5.67	**66.02**	**58.02**	**48.98**	**64.77**	**48.85**	23.36	**6.67**	-	**54.98**	**43.02**	**57.31**	**44.76**
	ETPNav (An et al., 2024)	-	5.03	-	61.46	50.83	66.41	51.28	-	5.64	-	54.79	44.89	61.90	45.33	-	6.99	-	51.21	39.86	54.11	41.30
	ETPNav*	18.16	5.06	64.06	62.09	50.64	66.06	51.17	18.92	5.96	63.66	54.83	44.62	61.36	44.87	21.83	6.92	-	51.38	39.90	53.85	40.91
	SC^2-WM	18.88	4.91	67.13	64.89	52.71	68.34	52.98	19.65	5.57	65.17	56.72	46.17	63.66	46.87	-	-	-	-	-	-	-
	HNR (Wang et al., 2024b)	-	4.85	-	63.72	53.17	68.81	52.78	-	5.51	-	56.39	46.73	63.56	47.24	-	6.81	-	53.22	41.14	55.61	42.89
	HNR*	19.74	4.93	66.01	63.55	53.37	69.02	52.66	20.41	5.75	64.93	56.48	46.62	63.43	47.38	23.02	6.88	-	53.33	41.18	55.47	42.95
	SC2-WM	20.54	4.98	**68.76**	65.10	**54.67**	**71.32**	**54.94**	21.01	5.72	**66.77**	**60.02**	**49.79**	**66.77**	**49.50**	-	-	-	-	-	-	-

Note: The official evaluation server for the Test Unseen split of RxR-CE dataset is currently unavailable, thus we only report results on the Val Seen and Unseen split.

Table 6. Ablation study of different loss functions for online model adaptation, including average time per episode. Best results are highlighted in **bold**.

Methods	R2R-CE Val Unseen							Time (s)
	TL ↓	NE ↓	OSR	SR	SPL	NDTW	SDTW	
SC^2-WM *w/o* Adaptation	18.73	5.46	59.22	49.86	36.05	48.10	34.62	17.78
SC^2-WM *w/* Self-Entropy Loss	18.62	5.46	58.62	49.48	35.84	48.32	34.52	19.93
SC^2-WM *w/* Async Scoring Loss	18.57	5.42	**59.43**	50.19	36.20	48.31	34.78	18.76
SC^2-WM *w/* FRL (Ours)	**17.65**	**5.37**	58.84	**50.90**	**37.17**	**49.31**	**35.70**	18.92

visual features from preceding steps to augment the current observation. We evaluate memory lengths $L \in \{2, 4, 6, 8\}$ and find that $L = 4$ yields the best performance, with results shown in Table 7. Shorter memory lengths (e.g., $L = 2$) provide insufficient historical context, leading to a 'myopic' agent that fails to capture necessary temporal dependencies. Conversely, excessive memory lengths (e.g., $L = 8$) degrade performance due to attention dilution. As the search space expands, the attention mechanism struggles to allocate focus efficiently, resulting in a dispersed (near-uniform) weight distribution that drowns out salient features. Thus, $L = 4$ strikes an optimal balance, providing rich context without overwhelming the attention module.

E. Real-World Verification

We further evaluate SC^2-WM on a real-world vision-and-language navigation platform using a Unitree GO2 quadruped robot equipped with an Intel RealSense D435i camera, as illustrated in Figure 5. Experiments are conducted in indoor home environments, covering an area of approximately $100m^2$ and includeing diverse furniture layouts/rooms. The robot is required to follow natural language navigation instructions such as "*Go through the table and find the trash can*". The robot executes actions (forward, turn left/right, stop) consistent with our simulation setup, and visual observations are processed in real-time. Depth sensing is utilized only for collision avoidance as a safety mechanism.

We perform 100 navigation trials across diverse layouts and instructions with varying levels of complexity. A trial is considered successful if the robot reaches the goal location described in the instruction (within 1.5 meters). SC^2-WM achieves an overall success rate of 85%, compared to 70% for the baseline. The performance gain is particularly notable in scenarios involving visual ambiguity or decision-making at intersections, where the self-correction mechanism enables the robot to recover from suboptimal actions. These results confirm that our approach transfers effectively from simulation to real-world deployment, demonstrating its practical applicability for VLN tasks.

Table 7. Ablation study of different memory lengths in the proposed World Model.

Methods	R2R-CE Val Unseen						
	TL ↓	NE ↓	OSR	SR	SPL	NDTW	SDTW
SC^2-WM ($L = 2$)	21.47	5.96	57.31	46.49	32.24	43.29	31.17
SC^2-WM ($L = 4$)	17.65	**5.37**	**58.84**	**50.90**	**37.17**	**49.31**	**35.70**
SC^2-WM ($L = 6$)	18.50	5.97	53.83	46.00	32.81	45.82	32.59
SC^2-WM ($L = 8$)	**16.13**	5.82	51.17	44.48	33.78	48.84	32.75

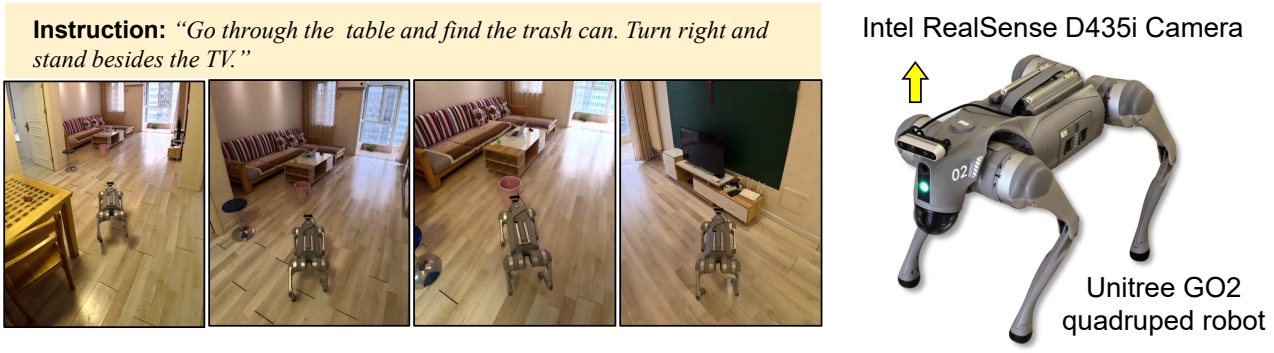

Figure 5. Real-world deployment of SC^2-WM on a quadruped robot navigating indoor environments with natural language instructions.