Path-level Hindsight Instructions for Semantic Exploration in Vision-Language Navigation

Sung June Kim[1,2⋆], Sangpil Kim[1†], and Honglak Lee[2†]

[1] Korea University, Seoul, South Korea
[2] University of Michigan, Ann Arbor, Michigan, USA

Abstract. On-policy exploration is a crucial component for training robust Vision-Language Navigation (VLN) agents, as it exposes the policy to a broader state distribution. However, such exploration inevitably leads to trajectories that deviate from expert demonstrations, resulting in a semantic mismatch between the executed visual stream and the original language instruction. In this work, we address this challenge by introducing Φ-Nav, a unified on-policy framework that leverages hindsight reasoning to align instructions with the agent's actual exploratory journey. Specifically, Φ-Nav operates through a three-stage dual-supervision cycle: 1) the agent performs oracle-guided on-policy exploration, sampling a trajectory while learning from expert action feedback, 2) a hindsight speaker synthesizes a path-level hindsight instruction grounded in the collected visual observations, and 3) the agent conducts a second imitation pass, treating the synthesized trajectory–instruction pair as an additional expert demonstration. Through this process, Φ-Nav bridges the critical semantic supervision gap inherent in on-policy methods, transforming semantically unlabeled movement into dense training signals. Evaluations on the R2R-CE and RxR-CE benchmarks show that Φ-Nav yields competitive performance while requiring only a fraction of the expert demonstrations used by current baselines. These results underscore the necessity of semantic exploration in VLN, positioning Φ-Nav as a effective solution for training embodied agents with limited data.

Keywords: Vision-language navigation · On-policy imitation learning · Hindsight experiential learning

1 Introduction

The field of Vision-Language Navigation (VLN) is a cornerstone of embodied AI, requiring agents to ground natural language instructions in complex, previously unseen environments [3, 59]. To achieve robust generalization, modern VLN systems increasingly rely on on-policy exploration during training, exposing agents to diverse and dynamically evolving state distributions [2, 13, 32, 37, 62]. While architectural advances have significantly improved perception and cross-modal

⋆ Work done at the University of Michigan as a visiting researcher.
† Corresponing authors {spk7@korea.ac.kr; honglak@umich.edu}

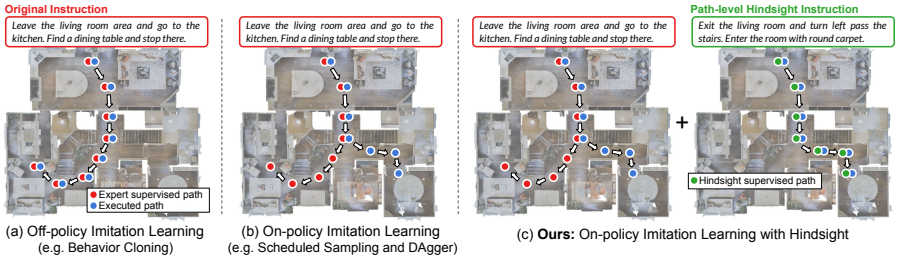

Fig. 1: Conceptual comparison of VLN training strategies. (a) Off-policy IL relies on static expert paths, failing to account for agent exploration. (b) On-policy IL exposes the agent to exploratory states but remains semantically limited to the original instruction, creating a supervision mismatch. (c) We utilize path-level hindsight relabeling to generate new instructions for exploratory trajectories, transforming deviations into meaningful training signals.

reasoning, the fundamental challenge of aligning exploratory behavior with coherent linguistic supervision remains unresolved.

To encourage robustness, many VLN policies adopt on-policy strategies such as scheduled sampling [6] and DAgger [45], which mitigate exposure bias by allowing agents to act under their own policies while receiving expert action feedback [1]. Although these methods broaden the explored state space, they remain semantically tethered to static, pre-collected instructions. For example, when an agent deviates from the expert's intended route, the original instruction no longer faithfully describes the executed visual stream. As a result, a substantial portion of exploratory trajectories lacks semantically aligned linguistic supervision, thereby limiting the full utilization of on-policy trajectories.

To bridge the semantic supervision gap in on-policy VLN policy training, we propose Φ(phi)-Nav, a framework that learns from path-level hindsight instructions. Our approach recognizes that every exploratory rollout contains a learnable narrative that remains invisible to traditional training objectives. To capture this latent knowledge, Φ-Nav introduces a three-stage dual-supervision cycle that functions as a wrapper around existing on-policy algorithms. First, the agent samples a trajectory while receiving real-time action corrections from an expert oracle. Next, Φ-Nav utilizes the robust multimodal spatio-temporal reasoning capability of pre-trained large vision-language models (LVLM) [5, 25] to retrospectively relabel its exploratory journey. By synthesizing path-level hindsight instructions that precisely describe the agent's actual visual observations, Φ-Nav transform arbitrary exploratory noise into dense, linguistically-grounded training demonstrations. Lastly, the agent performs a second imitation pass, treating the

[1] In VLN, Scheduled Sampling and DAgger are often used interchangeably to describe online expert querying with a decaying student-action probability. Strictly, DAgger aggregates data with iterative retraining, whereas Scheduled Sampling performs on-the-fly policy mixing without dataset aggregation. In this work, we use the latter to denote the online variant of DAgger.

synthesized trajectory–instruction pair as an additional expert demonstration, thereby reinforcing semantic grounding along the executed path.

Implementing hindsight relabeling for VLN presents unique challenges: unlike traditional hindsight paradigms that typically relabel final states within predefined goal space [4], our framework must narrate temporally-extended, open-ended trajectories, while at the same time mitigating LVLM hallucinations [34, 58]. We address these hurdles through two key mechanisms. First, we propose expert-in-context learning [7] to ensure synthesized instructions maintain structural and stylistic consistency with the training distribution. This enforces distributional alignment in linguistic style while allowing the specific visual observation to dictate the semantic content. Second, we introduce a trajectory–instruction alignment weighting mechanism [21, 46, 49] that adaptively weights hindsight supervision according to semantic fidelity. By evaluating the consistency between the visual trajectory and the synthesized instruction, the framework suppresses hallucinated or weakly grounded signals, ensuring that only reliable linguistic supervision influences policy optimization. Together, these components transform Φ-Nav into self-correcting, autonomous supervision engine that significantly enhances sample efficiency.

We conduct extensive experiments on the R2R-CE and RxR-CE dataset, and demonstrate distinctive benefits of Φ-Nav. Specifically, Φ-Nav not only improves the performance of current on-policy training when utilizing the full expert dataset, but also remains highly competitive while requiring fewer expert demonstrations. This advantage suggests that our path-level hindsight instructions provide a richer supervision signal than traditional fixed-trajectory demonstrations alone, paving the way for autonomous agents that can effectively self-supervise and scale their intelligence through direct environmental experience.

The contributions of this work can be summarized as follows:

- We propose a novel on-policy VLN framework Φ-Nav that bridges the missing semantic supervision gap in training by converting exploratory rollouts into dense path-level hindsight instructions.
- We extend hindsight paradigms to temporally-continuous trajectories, utilizing expert-in-context learning for distributional alignment and trajectory-instruction alignment weighting to ensure accurate, hallucination-free hindsight supervision.
- Evaluations on R2R-CE and RxR-CE demonstrate that Φ-Nav is sample-efficient, achieving competitive navigation performance while reducing dependence on expert data.

2 Related Work

2.1 Vision-Language Navigation

Vision-Language Navigation (VLN) is a fundamental task in embodied AI, requiring agents to ground and follow natural language instructions in 3D environments [3, 10, 28, 29, 51, 59]. The sequential decision making nature of the VLN

task opts for policy training strategies such as reinforcement learning (RL) or imitation learning (IL). RL-based methods [8, 11, 43, 54, 63] optimize navigation policies by learning to maximize a predefined reward/score function. However, due to the innate difficulties of RL in reward modeling, many VLN policies utilize IL as their main objectives [1–3, 12, 19, 27, 31, 57]. These methods utilize on-policy trajectory sampling methods such as scheduled sampling [6] and DAgger [45] since they effectively mitigate exposure bias by forcing the agent to learn from its own mistakes, thereby narrowing the distribution shift between training and inference. For instance, the foundational approach in [3] introduced student-forcing to the VLN domain, demonstrating that training on the agent's own sampled actions is essential for learning error-recovery behaviors.

Despite these advancements, a fundamental bottleneck remains: when an agent explores paths far from the expert demonstration, the original instruction becomes semantically irrelevant, leading to a semantic supervision gap. Our proposed Φ-Nav bridges this gap by generating path-level hindsight instructions that provide precise, dense supervision for any exploratory action the agent executes, effectively scaling training beyond the limits of static expert data.

2.2 Navigation Instruction Generation

Navigation Instruction Generation is an important sub-task in the VLN research, which enables precise and interpretable human-robot interaction. The Speaker-Follower framework [18] pioneered the "Speaker" model to back-translate paths into instructions, allowing for the creation of vast amounts of synthetic training data. This concept was further refined by EnvDrop [52], which performs environmental dropout to synthesize navigation pairs and significantly improve generalization. Marky [53] further scaled the augmentation scheme by extracting visual landmarks. Recent works leverage the vast pretrained knowledge of LVLMs, either utilizing them as a zero-shot generator [60, 64] or further fine-tuning them to generate linguistically diverse and spatially grounded instructions [15, 16, 30].

Generating hindsight instructions as in Φ-Nav poses several distinct challenges that separate its purpose from these works. While previous works focus on pretraining or offline augmentation, our framework is designed for providing semantic supervision for exploratory paths within on-policy training stream, which is a overlooked paradigm in existing literature. Furthermore, since on-policy exploration often yields suboptimal loops or deviations, Φ-Nav must rigorously verify the instruction stream. This ensures that only semantically- and logically-consistent labels are utilized to supervise the navigation policy.

2.3 Hindsight Experience Learning

Hindsight experience learning was introduced to improve sample efficiency in sparse-reward settings by relabeling failed trajectories with alternative achieved goals. The seminal Hindsight Experience Replay (HER) [4] introduced goal substitution to improve sample efficiency. Subsequent works extended hindsight relabeling to visual goal-conditioned settings [40, 42], hierarchical/multi-goal RL [33, 39], model-based imagination [20, 26], and curriculum/prioritized relabeling [17, 38]. More recently, inspired by hindsight principles, language models

have been used to provide retrospective feedback and self-refinement signals for policy learning [24, 50, 61, 65].

THER [14] and HSL [35] are closely related to our work in their use of language-guided hindsight signals, yet these methods mainly operate over structured predefined goal spaces, typically relabeling terminal states or scalar rewards. In contrast, Φ-Nav extends hindsight learning from goal substitution to temporally coherent, fine-grained path-level instruction relabeling. By integrating LVLM-based instruction synthesis with semantic verification, our framework establishes a hindsight paradigm tailored to embodied language grounding in on-policy VLN training.

3 Method

In this section, we detail the architecture and training paradigm of Φ-Nav. We first define the navigation environment and our exploration strategy in Section 3.1. Next, we detail the three stages of our pipeline: on-policy trajectory sampling (Section 3.2), path-level hindsight instruction generation (Section 3.3), and the final dual-supervision optimization process (Section 3.4). The overall framework of Φ-Nav is illustrated in Figure 2.

3.1 Problem Formulation

Vision-Language Navigation. In the Vision-Language Navigation (VLN) task, the agent must traverse a 3D environment to reach a destination specified by a natural language instruction I. At each discrete time step t, the agent receives a visual observation v_t, either in monocular or panoramic view. Based on the current visual observation and the instruction, the agent predicts next action $a_t \in \mathcal{A}$, where $\mathcal{A}$ is the total action space. The primary objective is to learn a policy $\pi_\theta(a_t|v_t, I)$ that maximizes the ratio of successfully reaching the goal while ensuring that every navigational action is precisely grounded in the semantic constraints of the instruction.

On-policy Exploration Strategies. To enable robust navigation, VLN policies frequently employ on-policy exploration strategies such as Scheduled Sampling [6] and DAgger [45] to expose the agent to a broader range of the state space. At each time step t, the action a_t is sampled according to:

$$a_t \sim \begin{cases} a_t^* & \text{w. p. } \epsilon^{(i)} \\ \pi_\theta(a|v_t, I_E^*) & \text{w. p. } 1-\epsilon^{(i)}, \end{cases}$$

where a_t^* denotes the expert action and $\epsilon^{(i)}$ is a decay factor for the i-th training episode that gradually shifts the sampling from the expert's teacher-forcing to the agent's current policy π_θ. While these methods effectively expand the visited state space, they introduce a significant semantic gap. When the agent samples

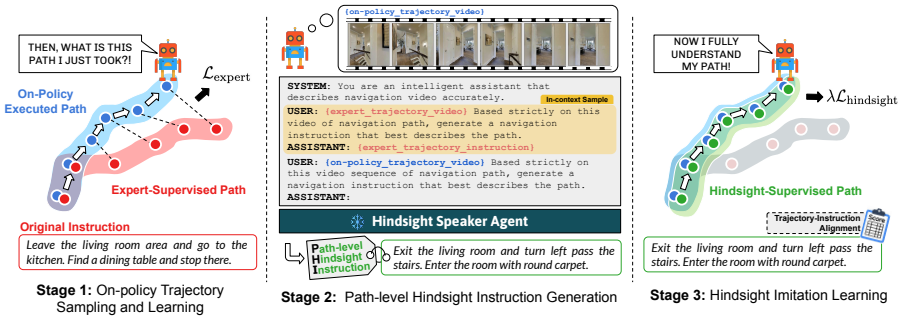

Fig. 2: Overview of the Φ-Nav pipeline. In Stage 1, the agent samples a trajectory while being supervised by expert actions. In Stage 2, the agent analyzes its path in hindsight to generate a verified instruction that aligns with the observed visual stream. Lastly, in Stage 3, the agent re-evaluates its past decision points under the guidance of this new instruction, learning to ground its actions with semantic supervision that aligns. Full prompt is presented in the Appendix.

$a_t \neq a_t^*$, it generates an exploratory trajectory $\tau = \{v_0, a_0, \ldots, v_T, a_T\}$ that deviates from the expert path τ_E^*, making the instruction for the expert trajectory I_E^* semantically unmatched with the resulting visual stream. To address these deviations, Φ-Nav introduces a hindsight mechanism in VLN by synthesizing instructions that accurately reflect the agent's actual exploratory paths. By providing valid linguistic supervision for the path actually executed, Φ-Nav bridges the gap between exploration and semantic instruction grounding.

3.2 Stage 1: On-policy Trajectory Sampling and Learning

Φ-Nav begins with the standard training procedure adopted by prior VLN policies [2, 12, 31]. At each time step t, the agent processes its visual observation and the original language instruction to generate a probability distribution of possible actions using policy π_θ. Then, following the sampling strategies in Section 3.1, the navigation agent initiates exploration to collect on-policy trajectory $\tau = \{v_0, a_0, \ldots, v_T, a_T\}$. Regardless of the selected actions, the agent's own predictions are supervised and corrected by the offline expert action a^*, by learning to minimize the cross-entropy loss:

$$\mathcal{L}_{\text{expert}} = -\sum_{t=0}^{T} \log \pi_\theta(a_t^* \mid v_t, I_E^*). \tag{1}$$

Through this supervision, the agent learns how to mimic the expert's decision-making process within the observed states. However, most works stop here, leaving the semantic supervision for exploratory, on-policy trajectories unaddressed, as the instruction for the expert trajectory I_E^* may become misaligned with the agent's executed trajectory. Φ-Nav effectively bridges this gap through the following stages.

3.3 Stage 2: Path-level Hindsight Instruction Generation

The core of Φ-Nav lies in its ability to transform raw, exploratory visual streams into semantically-grounded language instruction. Once a trajectory τ is buffered from Stage 1, we employ a large vision-language model (LVLM) based Hindsight Speaker Agent with expert-in-context learning strategy to synthesize a reliable hindsight instruction I_H.

Hindsight Speaker Agent. We leverage the extensive zero-shot spatio-temporal reasoning and linguistic knowledge of LVLMs to act as the Hindsight Speaker Agent (HSA). The advantage of zero-shot LVLMs over task-specific fine-tuned speakers [16, 30] lies in their stronger generalization and linguistic flexibility under distribution shift. While fine-tuned speakers are typically trained to reconstruct instructions conditioned on ideal ground-truth paths, zero-shot HSA can generate visually grounded descriptions even when the agent substantially deviates from the expert route, which is a common scenario for on-policy trajectories. For instance, if the agent takes a wrong turn and traverses a visual state rarely observed in training demonstrations, a fine-tuned speaker may struggle to produce a coherent or accurate instruction due to its reliance on expert-aligned data. In contrast, HSA can still generate a faithful description of the detour, thereby maintaining semantic alignment with the executed trajectory.

Expert-In-Context Learning. To further stabilize hindsight instruction generation, we incorporate an expert-in-context learning strategy. Although HSA could operate in a zero-shot manner, we provide a single exemplar consisting of an expert trajectory and its corresponding human-annotated instruction. Specifically, for each generation, we sample a trajectory–instruction pair:

$$(\tau^*, I^*) \sim p_{\mathcal{E}},$$

where $p_{\mathcal{E}}$ denotes the empirical distribution induced by the offline expert demonstration set $\mathcal{E} = \{(\tau_i^*, I_i^*)\}_{i=1}^N$. Pure zero-shot generation may introduce excessive diversity or creativity in phrasing, which can induce distribution shift and hinder policy learning. By contrast, conditioning on an expert demonstration encourages HSA to produce instructions that resemble human-authored navigation commands in tone, structure, and granularity, while still grounding the semantic content in the visual observations of the current on-policy trajectory.

3.4 Stage 3: Hindsight Imitation Learning

Once the path-level hindsight instruction is generated, Φ-Nav conducts a second round of imitation learning using the synthesized trajectory–instruction pair as augmented supervision. However, because the instruction is produced by an LVLM, it may contain imperfect or weakly grounded descriptions. To mitigate potential noise, we introduce a trajectory–instruction alignment score that weights the contribution of hindsight supervision during policy optimization.

Trajectory-Instruction Alignment Weighting. Since Φ-Nav operates within the on-policy training stream, the alignment between a trajectory and its hindsight instruction must be evaluated online and without ground-truth references. To this end, we design a lightweight trajectory–instruction alignment scoring module inspired by EMScore [49], which evaluates video–caption consistency via hierarchical embedding similarities: a coarse score computed from global video and sentence embeddings, and a fine-grained score from frame–word similarity.

While this hierarchical design is well suited for generic video captioning, navigation scenarios place particular emphasis on landmark grounding, since they serve as critical spatial anchors for decision making. Therefore, instead of computing fine-grained similarity over all words, we focus specifically on landmark nouns in the instruction and measure their alignment with individual trajectory frames. First, we compute the coarse alignment score as a cosine similarity between global trajectory video and sentence embeddings:

$$\mathcal{S}_{\text{coarse}} = \text{sim}\Big(\frac{1}{|V|}\sum_{v_i \in V} f(v_i),\ g(I_H)\Big), \tag{2}$$

where V denotes the whole video frames and f and g are the CLIP [44] image and text encoder, respectively. Next, for fine-grained alignment, we first extract landmark nouns from I_H using spaCy [23] and obtain a set of landmark nouns M. We then compute bidirectional frame–landmark alignment scores:

$$A_{M\to V} = \frac{1}{|M|}\sum_{m_j \in M} \max_{v_i \in V} \text{sim}\big(f(v_i),\ g(m_j)\big) \tag{3}$$

$$A_{V\to M} = \frac{1}{|V|}\sum_{v_i \in V} \max_{m_j \in M} \text{sim}\big(f(v_i),\ g(m_j)\big), \tag{4}$$

and define the fine-grained score as their harmonic mean:

$$\mathcal{S}_{\text{fine}} = \frac{2 \cdot A^+_{M\to V} \cdot A^+_{V\to M}}{A^+_{M\to V} + A^+_{V\to M} + \epsilon}, \tag{5}$$

where $A^+ = \max(0, A)$ ensures non-negative alignment contributions, and $\epsilon > 0$ is a small constant to prevent division by zero. We define the trajectory-instruction alignment weight λ as the average of $\mathcal{S}_{\text{coarse}}$ and $\mathcal{S}_{\text{fine}}$, which adaptively modulates the contribution of hindsight signals in each episode. While more sophisticated online reference-free scoring strategies may be explored, we adopt this lightweight formulation for efficiency and leave further improvements to future work.

Hindsight Loss Function. In the final stage of the Φ-Nav cycle, the agent updates its policy by treating its own exploratory journey as a successful execution of the newly synthesized instruction. Given a completed on-policy trajectory $\tau = \{v_0, a_0, \ldots, v_T, a_T\}$ and its corresponding hindsight instruction I_H, the agent

is supervised to maximize the likelihood of the actions it actually performed. The hindsight imitation loss is formulated as:

$$\mathcal{L}_{\text{hindsight}} = -\sum_{t=0}^{T} \log \pi_\theta(a_t \mid v_t, I_H). \tag{6}$$

Crucially, this objective allows the agent to learn the relationship between linguistic cues and visual state transitions even in regions of the environment far from the expert distribution. To balance traditional imitation learning with this self-supervised signal, we combine the expert loss from Eq. 1 with the weighted hindsight loss. The total optimization objective is defined as:

$$\mathcal{L} = \mathcal{L}_{\text{expert}} + \lambda \mathcal{L}_{\text{hindsight}} \tag{7}$$

where λ is the weighting coefficient derived from the trajectory–instruction alignment scores. Through this optimization, Φ-Nav ensures that every navigation episode, whether successful or exploratory, contributes to a more robust and semantically-aware on-policy training.

4 Experiment

4.1 Experiment Setup

Datasets. Φ-Nav is trained and evaluated on the R2R-CE and RxR-CE datasets. These datasets provide expert navigation trajectories paired with natural language instructions. The visual observations along each trajectory are rendered using the Habitat Simulator [47], enabling photo-realistic embodied navigation in continuous 3D environments. The R2R-CE dataset consists of 5,611 trajectories divided into train, validation seen, validation unseen, and test unseen [2]. Each trajectory is annotated with three English instructions. The dataset features an average path length of 9.89 meters and an average instruction length of 32 words. The RxR-CE dataset provides navigation instructions in three languages—English, Hindi, and Telugu—with substantially longer descriptions averaging approximately 120 words per instruction. In addition, RxR-CE features longer trajectories on average than R2R-CE, increasing the difficulty of long-horizon reasoning and fine-grained instruction grounding.

Evaluation Metrics. For navigation performances, we adopt the following metrics. Navigation Error (NE): average distance in metric space between the final and target location; Success Rate (SR): the ratio of episodes with NE less than 3.0 meters; Oracle SR (OSR): SR given the oracle stop policy; SR penalized by Path Length (SPL), Trajectory Length (TL): average path length in meters; Normalize Dynamic Time Wrapping (NDTW): the similarity between

[2] Due to the deprecation of the official VLN-CE evaluation server on Eval.ai, we were unable to obtain test-unseen results.

the predicted and expert paths and lastly NDTW penalized by SR (SDTW). For instruction generation, we report the BLEU [41] and ROUGE-L [36] metric in Section 4.3,which quantify lexical overlap and sequence-level similarity between generated instructions and the ground-truth references.

Baselines and Implementation Details. We evaluate the applicability of Φ-Nav under two on-policy imitation learning paradigms: DAgger [45] and scheduled sampling [6]. For the DAgger setting, we adopt CMA [31] as the baseline policy. CMA is a representative baseline in the VLN-CE benchmark, designed to model multimodal grounding via cross-modal attention and temporal dependencies through recurrent neural networks. Specifically, we use the CMA-DA and CMA-DA-PM-Aug variants, which are the DAgger-trained versions of CMA. For these experiments, we follow the DAgger sample collection strategy and store the hindsight-generated trajectory–instruction pairs in the same buffer for subsequent training. Next, for the scheduled sampling setting, we adopt ETPNav [2], which leverages a topological map for structured decision-making. Here, the policy updates are performed on a per-sample basis. Accordingly, Φ-Nav is applied in an iterative three-stage cycle for each sampled trajectory.

The Hindsight Speaker Agent is implemented using the Qwen2.5-VL-7B model [5] as backbone. We provide ablation on different backbone choices in Section 4.5. Additionally, we provide full prompt template and hyperparameters in the Appendix. For panoramic observation settings, we select four non-overlapping views centered around the front-facing direction and use them as input frames to the Hindsight Speaker Agent. The rest of the training configuration adhere to the default setting of the baseline policies. The experiments were conducted on NVIDIA L40S GPUs.

4.2 Main Navigation Results

Evaluation on R2R-CE. The results in Table 1 indicate that incorporating additional semantic exploration through Φ-Nav consistently improves performance under on-policy training paradigms. For example, in DAgger-based experiments, Φ-Nav improves the SR of the CMA-D-P-A baseline by 1.98 percentage points on the val unseen split. Similarly, in scheduled sampling based experiments, Φ-Nav improves the SR and the SPL of ETPNav by 5.37 and 3.59 percentage points in the val unseen split, respectively. It is also worth noting that although ETPNav alone achieves lower performance, integrating Φ-Nav improves it to slightly surpass the recent state-of-the-art method g3D-LF [55]. These results clearly highlight the necessity of filling the missing semantic supervision gap in on-policy explorations.

Evaluation on RxR-CE. Consistent with the results on R2R-CE, Table 2 shows that Φ-Nav achieves stable numerical improvements across both validation splits. In particular, on the val unseen split, Φ-Nav improves SR and SPL by 1.04 and 1.06 percentage points, respectively. Additionally, gains in the DTW-based

Table 1: Experimental results on the R2R-CE dataset. Performance improvements obtained by applying Φ-Nav are highlighted in bold. Methods marked with † use monocular-view observations.

Methods	Val Seen					Val Unseen				
DAgger [45]	TL	NE↓	OSR↑	SR↑	SPL↑	TL	NE↓	OSR↑	SR↑	SPL↑
CMA-D† [31]	8.64	6.79	41.77	32.87	30.28	7.86	8.17	33.22	26.64	24.88
w/ Φ-Nav	8.32	**6.71**	**43.44**	**34.83**	**33.13**	7.89	**7.69**	**35.12**	**27.62**	**25.92**
CMA-D-P-A† [31]	9.63	7.11	46.13	37.17	34.82	8.86	7.62	40.28	32.10	29.92
w/ Φ-Nav	9.54	**6.99**	**47.18**	**38.80**	**35.63**	8.27	**7.42**	**42.23**	**34.08**	**31.51**
Sched. Sampling [6]	TL	NE↓	OSR↑	SR↑	SPL↑	TL	NE↓	OSR↑	SR↑	SPL↑
VLN-BERT [22]	12.50	5.02	59	50	44	12.23	5.74	53	44	39
BEVBert [1]	13.98	3.77	73	68	60	-	4.57	67	59	50
HNR [56]	11.79	3.67	76	69	61	12.64	4.42	67	61	51
ENP-ETPNav [37]	11.82	3.90	73	68	59	11.45	4.69	65	58	50
g3D-LF [55]	11.61	3.72	72	63	55	-	4.53	68	61	52
ETPNav [2]	11.38	3.94	72.23	66.45	59.62	11.99	4.71	64.71	57.21	49.15
w/ Φ-Nav	11.29	**3.38**	**75.06**	**68.63**	**61.41**	12.76	**4.29**	**68.84**	**62.58**	**52.74**

Table 2: Experimental results on the RxR-CE dataset. Performance improvements obtained by applying Φ-Nav are highlighted in bold.

Methods	Val Seen					Val Unseen				
Sched. Sampling [6]	NE↓	SR↑	SPL↑	NDTW↑	SDTW↑	NE↓	SR↑	SPL↑	NDTW↑	SDTW↑
VLN-BERT [22]	-	-	-	-	-	8.98	27.08	22.65	46.71	-
HNR [56]	4.85	63.72	53.17	68.81	52.78	5.51	56.39	46.73	63.56	47.24
ENP-ETPNav [37]	5.10	62.01	51.18	67.22	51.90	5.51	55.27	45.11	62.97	45.83
ETPNav [2]	5.03	61.46	50.83	66.41	51.28	5.64	54.79	44.89	61.90	45.33
w/ Φ-Nav	**5.02**	**62.84**	**51.34**	**67.92**	**51.70**	5.67	**55.83**	**45.95**	**62.93**	**46.11**

metrics indicate improved trajectory fidelity even in the linguistically dense and long-horizon navigation scenarios of RxR-CE. However, the magnitude of improvement is smaller than on R2R-CE, likely because RxR-CE requires generating finer-grained instructions over substantially longer trajectories, making accurate hindsight instruction generation more challenging.

4.3 Instruction Generation Analysis

Trajectory-Instruction Alignment. In this section, we visually examine how the generated instructions are semantically grounded in the visual stream of the trajectories and how such grounding contributes to hindsight imitation learning, as quantified by the trajectory–instruction alignment weight (TIAW) [3]. In the first example of Figure 3, the on-policy trajectory diverges from the expert path at an early stage. However, Φ-Nav accurately captures the actual executed trajectory, introducing additional descriptive phrase (*e.g., large mirror on the wall*)

[3] The distribution of TIAW scores across trajectories is provided in the Appendix.

Fig. 3: Trajectory–Instruction Alignment. Generated hindsight instructions are shown in green boxes, with major landmarks highlighted in bold and underlined. The gray text below each corresponds to the original expert instruction.

that encourages richer visual grounding along the way. The final position aligns precisely with the *kitchen counter* stop signal, yielding a relatively high TIAW of 0.53. In the second example, the generated instruction accurately captures directional cues (*e.g., right, left*), as well as fine-grained visual details. Compared to the first example, however, many of the referenced landmarks occupy relatively small regions in the pixel space. This reduced perceptual saliency weakens the visual grounding signal, leading to slightly lower TIAW of 0.46 despite the instruction remaining semantically accurate. In the final example, the observations lack distinctive landmarks, hence the visual stream provides limited object-level anchors for semantic grounding. Consequently, the trajectory–instruction correspondence is substantially weaker, yielding a low TIAW of 0.22. In summary, Φ-Nav selectively emphasizes well-grounded trajectories and down-weights weakly aligned ones, enabling more stable and adaptive optimization.

Effect of Expert-In-Context Learning. Our proposed expert-in-context learning strategy constrains the open-ended language space to remain within the training instruction distribution for stable hindsight optimization. We validate this by comparing against a pure zero-shot setting and a template-based prompting strategy adopted in InstruGen [60], generating instructions on offline expert trajectories and measuring their lexical and structural similarity to the ground-truth instructions of R2R-CE. For these comparisons, we exclude the original instruction from each trajectory to avoid trivial matches. We use POS-tagging [23] for structural analysis. The results in Table 3 show that fixed

Table 3: Distributional alignment between ground-truth and generated instructions.

	Lexical			Structural			Nav.
Methods	BLEU-1↑	BLEU-4↑	ROUGE-L↑	BLEU-1↑	BLEU-4↑	ROUGE-L↑	SPL↑
Zero-shot	0.2510	0.022	0.2039	0.3724	0.1402	0.3492	50.55
Template-based [60]	0.1733	0.0193	0.1648	0.3201	0.0932	0.2723	-
Expert-In-Context	**0.3217**	**0.087**	**0.3192**	**0.5394**	**0.3129**	**0.5952**	**52.74**

Table 4: Mean TIAW scores with standard deviation

	Expert Reference	Zero-Shot	**Expert-In-Context**
TIAW	0.492 (± 0.06)	0.278 (± 0.07)	**0.307** (± 0.07)
TIAW†	0.517 (± 0.07)	0.291 (± 0.05)	**0.318** (± 0.06)

template-based generation evidently fails to align with the training distribution. In contrast, our approach preserves structural similarity with the ground-truth distribution, naturally leading to better navigation results when used for on-policy training.

Semantic Faithfulness of the Hindsight Supervision. The proposed TIAW quantifies the semantic fidelity of generated instructions, allowing the framework to dynamically weight the contribution of hindsight supervision based on its reliability. To systematically validate its faithfulness, we measured TIAW across three categories: ground-truth expert references for calibration, zero-shot based, and our Expert-In-Context Learning-based generations derived from on-policy rollouts. Table 4 reports the mean TIAW scores with standard deviation. First, TIAW correctly identifies expert references as having the highest semantic fidelity, establishing a clear upper bound for calibration. Furthermore, Expert-in-context consistently outperforms zero-shot generation in both standard and landmark-aware (TIAW†) variants. This provides direct evidence that distributional similarity highly correlates with semantic correspondence between the visual trajectory and the generated text. Altogether, this calibration validates TIAW as a principled filtering mechanism, ensuring that the hindsight supervision is semantically faithful.

4.4 Sample Efficiency Analysis

Unlike traditional methods that rely solely on the original expert instructions, Φ-Nav extracts dense learning signals from exploratory deviations, enabling sample efficient training in two perspectives. First, as illustrated in Figure 4-(a), Φ-Nav consistently outperforms the baseline across all stages of training. This is primarily due to Φ-Nav exposing the agent to a higher volume of state-instruction pairs per iteration. Consequently, Φ-Nav achieves higher success rates with fewer environmental interactions, proving that hindsight supervision is a more potent signal for policy optimization than expert-action correction alone. Second, Figure 4-(b) highlights Φ-Nav's ability to maintain high performance even when expert demonstrations are scarce. Specifically, we observe that Φ-Nav surpasses

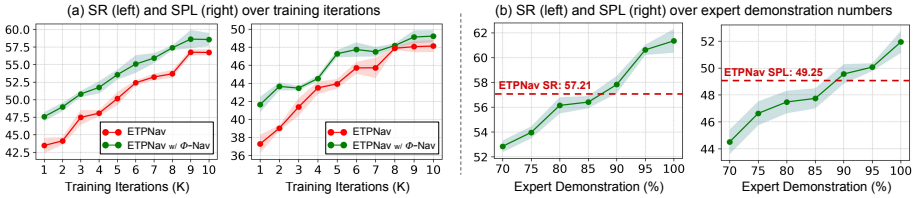

Fig. 4: Sample Efficiency Analysis. Results are averaged over three independent runs, with the shaded region indicating standard deviation.

Table 5: Navigation results using different weighting mechanisms. TIAW† uses landmark extraction.

Method	TL	NE↓	SR↑	SPL↑
$\lambda = 0.1$	13.12	4.94	59.12	48.66
$\lambda = 1.0$	13.57	5.06	58.58	46.32
TIAW	12.91	4.46	61.18	51.32
TIAW†	12.76	**4.29**	**62.58**	**52.74**

Table 6: Navigation results using LVLM backbones in different sizes.

Method	TL	NE↓	SR↑	SPL↑
Qwen2.5-VL-3B	13.52	4.97	58.11	47.32
Qwen2.5-VL-7B	12.76	**4.29**	**62.58**	**52.74**
Qwen3-VL-8B	12.44	4.41	60.05	51.18

the baseline's best result using only 90% of the available expert data. By augmenting the training distribution with high-fidelity hindsight demonstrations, Φ-Nav enables the baseline model to achieve better performance with a reduced reliance on human-annotated expert data.

4.5 Ablation Studies

We provide ablation studies comparing different hindsight weighting mechanisms and LVLM backbones. For these experiments, we use ETPNav [2] as the baseline and report results on the val unseen split of the R2R-CE dataset.

Hindsight Weighting Mechanisms. In Table 5, setting the hindsight weight λ to a fixed scalar yields suboptimal performance compared to our Trajectory–Instruction Alignment Weighting (TIAW) variants, which adaptively controls the contribution of each sample. Among the fixed-weight settings, smaller λ values perform better, suggesting that reducing hindsight signals mitigates learning from noisy or hallucinated supervision. For the TIAW variant, using the landmark-based scoring consistently outperforms the variant without it across NE, SR, and SPL, indicating that landmark-aware alignment yields more reliable and semantically grounded weighting.

LVLM Backbones. Table 6 reports navigation performance using different LVLM backbones from the Qwen-VL series [5], comparing models of varying sizes. We observe a clear performance gap between the 3B model and the larger variants. Qwen2.5-VL-3B yields the weakest results across all metrics, suggesting that limited model capacity constrains the quality of generated hindsight instructions and downstream policy learning. Comparing Qwen2.5-VL-7B and Qwen3-VL-8B, both larger models achieve substantial improvements over the 3B variant.

While Qwen3-VL-8B attains slightly shorter trajectory lengths, Qwen2.5-VL-7B achieves the best overall navigation performance. These results indicate that increasing model scale generally benefits hindsight instruction generation, but architectural differences and alignment characteristics also play a critical role beyond parameter count alone.

5 Conclusion

We introduced Φ-Nav, a novel on-policy VLN training framework that addresses the semantic supervision gap arising in exploratory rollouts. Specifically, Φ-Nav converts misaligned trajectories into semantically grounded path-level hindsight instructions, by leveraging the spatio-temporal reasoning of pre-trained LVLMs. Furthermore, to ensure that hindsight supervision remains both faithful and distribution-consistent, we devise expert-in-context learning for structural alignment and trajectory–instruction alignment weighting for adaptive optimization. Experiments on R2R-CE and RxR-CE demonstrate that Φ-Nav not only strengthens standard on-policy imitation learning but also substantially improves sample efficiency, reducing reliance on costly expert demonstrations. Overall, Φ-Nav demonstrates the potential of language-guided self-supervision to advance embodied AI, enabling agents to learn to make decisions from their own experiences.

Limitations and Future Work. Although our strategies improve instruction generation from LVLMs, the outputs remain imperfect. The trajectory–instruction alignment weight λ could also benefit from more sophisticated online, reference-free scoring. Future work will focus on improving generation efficiency, alignment estimation, and extending the framework to longer-horizon and broader robotic VLA tasks.

Acknowledgement

This work was supported by Culture, Sports and Tourism R&D Program through the Korea Creative Content Agency grant funded by the Ministry of Culture, Sports and Tourism (International Collaborative Research and Global Talent Development for the Development of Copyright Management and Protection Technologies for Generative AI, RS-2024-00345025), the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(RS-2025-00521602), Institute of Information & communications Technology Planning & Evaluation (IITP) & ITRC(Information Technology Research Center) grant funded by the Korea government(MSIT) (No.RS-2019-II190079, Artificial Intelligence Graduate School Program(Korea University), 1%);

References

1. An, D., Qi, Y., Li, Y., Huang, Y., Wang, L., Tan, T., Shao, J.: Bevbert: Multimodal map pre-training for language-guided navigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2023)

2. An, D., Wang, H., Wang, W., Wang, Z., Huang, Y., He, K., Wang, L.: Etpnav: Evolving topological planning for vision-language navigation in continuous environments. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
3. Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., Van Den Hengel, A.: Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3674–3683 (2018)
4. Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Pieter Abbeel, O., Zaremba, W.: Hindsight experience replay. Advances in neural information processing systems **30** (2017)
5. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)
6. Bengio, S., Vinyals, O., Jaitly, N., Shazeer, N.: Scheduled sampling for sequence prediction with recurrent neural networks. Advances in neural information processing systems **28** (2015)
7. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
8. Bundele, V., Bhupati, M., Banerjee, B., Grover, A.: Scaling vision-and-language navigation with offline rl. arXiv preprint arXiv:2403.18454 (2024)
9. Chaplot, D.S., Gandhi, D.P., Gupta, A., Salakhutdinov, R.R.: Object goal navigation using goal-oriented semantic exploration. Advances in Neural Information Processing Systems **33**, 4247–4258 (2020)
10. Chen, K., An, D., Huang, Y., Xu, R., Su, Y., Ling, Y., Reid, I., Wang, L.: Constraint-aware zero-shot vision-language navigation in continuous environments. IEEE Transactions on Pattern Analysis and Machine Intelligence (2025)
11. Chen, S., Guhur, P.L., Schmid, C., Laptev, I.: History aware multimodal transformer for vision-and-language navigation. Advances in neural information processing systems **34**, 5834–5847 (2021)
12. Chen, S., Guhur, P.L., Tapaswi, M., Schmid, C., Laptev, I.: Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16537–16547 (2022)
13. Cheng, A.C., Ji, Y., Yang, Z., Gongye, Z., Zou, X., Kautz, J., Bıyık, E., Yin, H., Liu, S., Wang, X.: Navila: Legged robot vision-language-action model for navigation. arXiv preprint arXiv:2412.04453 (2024)
14. Cideron, G., Seurin, M., Strub, F., Pietquin, O.: Self-educated language agent with hindsight experience replay for instruction following (2019)
15. Fan, S., Liu, R., Wang, W., Yang, Y.: Navigation instruction generation with bev perception and large language models. In: European Conference on Computer Vision. pp. 368–387. Springer (2024)
16. Fan, S., Liu, R., Wang, W., Yang, Y.: Scene map-based prompt tuning for navigation instruction generation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 6898–6908 (2025)
17. Fang, M., Zhou, T., Du, Y., Han, L., Zhang, Z.: Curriculum-guided hindsight experience replay. In: Advances in Neural Information Processing Systems (NeurIPS) (2019)

18. Fried, D., Hu, R., Cirik, V., Rohrbach, A., Andreas, J., Morency, L.P., Berg-Kirkpatrick, T., Saenko, K., Klein, D., Darrell, T.: Speaker-follower models for vision-and-language navigation. Advances in neural information processing systems **31** (2018)
19. Gao, J., Liu, R., Wang, W.: 3d gaussian map with open-set semantic grouping for vision-language navigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9252–9262 (2025)
20. Hafner, D., Lillicrap, T., Ba, J., Norouzi, M.: Dream to control: Learning behaviors by latent imagination. In: International Conference on Learning Representations (ICLR) (2020)
21. Hessel, J., Holtzman, A., Forbes, M., Le Bras, R., Choi, Y.: Clipscore: A reference-free evaluation metric for image captioning. In: Proceedings of the 2021 conference on empirical methods in natural language processing. pp. 7514–7528 (2021)
22. Hong, Y., Wang, Z., Wu, Q., Gould, S.: Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15439–15449 (2022)
23. Honnibal, M., Montani, I.: spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing (2017), to appear
24. Huang, W., Abbeel, P., Pathak, D., Mordatch, I.: Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In: International Conference on Machine Learning (ICML) (2022)
25. Hurst, A., Lerer, A., Goucher, A.P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al.: Gpt-4o system card. arXiv preprint arXiv:2410.21276 (2024)
26. Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R.H., Czechowski, K., Erhan, D., Finn, C., Kozakowski, P., Levine, S.: Model-based reinforcement learning for atari. In: International Conference on Learning Representations (ICLR) (2020)
27. Kamath, A., Anderson, P., Wang, S., Koh, J.Y., Ku, A., Waters, A., Yang, Y., Baldridge, J., Parekh, Z.: A new path: Scaling vision-and-language navigation with synthetic instructions and imitation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10813–10823 (2023)
28. Kim, S., Oh, G., Ko, H., Ji, D., Lee, D., Lee, B.J., Jang, S., Kim, S.: Test-time adaptation for online vision-language navigation with feedback-based reinforcement learning. In: Forty-second International Conference on Machine Learning (2025)
29. Ko, H., Kim, S., Oh, G., Yoon, J., Lee, H., Jang, S., Kim, S., Kim, S.: Active test-time vision-language navigation. arXiv preprint arXiv:2506.06630 (2025)
30. Kong, X., Chen, J., Wang, W., Su, H., Hu, X., Yang, Y., Liu, S.: Controllable navigation instruction generation with chain of thought prompting. In: European Conference on Computer Vision. pp. 37–54. Springer (2024)
31. Krantz, J., Wijmans, E., Majumdar, A., Batra, D., Lee, S.: Beyond the nav-graph: Vision-and-language navigation in continuous environments. In: European Conference on Computer Vision. pp. 104–120. Springer (2020)
32. Lee, J., Bjelonic, M., Reske, A., Wellhausen, L., Miki, T., Hutter, M.: Learning robust autonomous navigation and locomotion for wheeled-legged robots. Science Robotics **9**(89), eadi9641 (2024)

33. Levy, A., Konidaris, G., Platt, R., Saenko, K.: Hierarchical reinforcement learning with hindsight. In: International Conference on Learning Representations (ICLR) (2019)
34. Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W.X., Wen, J.R.: Evaluating object hallucination in large vision-language models. In: Proceedings of the 2023 conference on empirical methods in natural language processing. pp. 292–305 (2023)
35. Li, Z., Wu, G., Wang, Z., Morariu, V.I., Zhang, R., Zhu, W., Rossi, R.A., Kil, J.: Spinning straw into gold: Relabeling llm agent trajectories in hindsight for successful demonstrations. In: The Fourteenth International Conference on Learning Representations (2026)
36. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out. pp. 74–81 (2004)
37. Liu, R., Wang, W., Yang, Y.: Vision-language navigation with energy-based policy. Advances in Neural Information Processing Systems **37**, 108208–108230 (2024)
38. Luo, Y., Xu, Y., Liu, T., Zhang, Z.: Energy-based hindsight experience prioritization. In: AAAI Conference on Artificial Intelligence (2020)
39. Nachum, O., Gu, S., Lee, H., Levine, S.: Data-efficient hierarchical reinforcement learning. In: Advances in Neural Information Processing Systems (NeurIPS) (2018)
40. Nair, A., Pong, V., Dalal, M., Bahl, S., Lin, S., Levine, S.: Visual reinforcement learning with imagined goals. In: Advances in Neural Information Processing Systems (NeurIPS) (2018)
41. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics. pp. 311–318 (2002)
42. Pong, V., Gu, S., Dalal, M., Levine, S.: Temporal difference models: Model-free deep rl for model-based control. In: International Conference on Learning Representations (ICLR) (2018)
43. Qi, Z., Zhang, Z., Yu, Y., Wang, J., Zhao, H.: Vln-r1: Vision-language navigation via reinforcement fine-tuning. arXiv preprint arXiv:2506.17221 (2025)
44. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021)
45. Ross, S., Gordon, G., Bagnell, D.: A reduction of imitation learning and structured prediction to no-regret online learning. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. pp. 627–635. JMLR Workshop and Conference Proceedings (2011)
46. Sarto, S., Barraco, M., Cornia, M., Baraldi, L., Cucchiara, R.: Positive-augmented contrastive learning for image and video captioning evaluation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6914–6924 (2023)
47. Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., et al.: Habitat: A platform for embodied ai research. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9339–9347 (2019)
48. Shah, D., Eysenbach, B., Kahn, G., Rhinehart, N., Levine, S.: Ving: Learning open-world navigation with visual goals. In: 2021 IEEE International Conference on Robotics and Automation (ICRA). pp. 13215–13222. IEEE (2021)
49. Shi, Y., Yang, X., Xu, H., Yuan, C., Li, B., Hu, W., Zha, Z.J.: Emscore: Evaluating video captioning via coarse-grained and fine-grained embedding matching. In: Pro-

ceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 17929–17938 (2022)
50. Shinn, N., Labash, S., Gopinath, A., et al.: Reflexion: Language agents with verbal reinforcement learning. In: Advances in Neural Information Processing Systems (NeurIPS) (2023)
51. Song, X., Chen, W., Liu, Y., Chen, W., Li, G., Lin, L.: Towards long-horizon vision-language navigation: Platform, benchmark and method. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12078–12088 (2025)
52. Tan, H., Yu, L., Bansal, M.: Learning to navigate unseen environments: Back translation with environmental dropout. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 2610–2621 (2019)
53. Wang, S., Montgomery, C., Orbay, J., Birodkar, V., Faust, A., Gur, I., Jaques, N., Waters, A., Baldridge, J., Anderson, P.: Less is more: Generating grounded navigation instructions from landmarks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15428–15438 (2022)
54. Wang, X., Huang, Q., Celikyilmaz, A., Gao, J., Shen, D., Wang, Y.F., Wang, W.Y., Zhang, L.: Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6629–6638 (2019)
55. Wang, Z., Lee, G.H.: g3d-lf: Generalizable 3d-language feature fields for embodied tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14191–14202 (2025)
56. Wang, Z., Li, X., Yang, J., Liu, Y., Hu, J., Jiang, M., Jiang, S.: Lookahead exploration with neural radiance representation for continuous vision-language navigation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13753–13762 (2024)
57. 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 (2025)
58. Wu, T.H., Lee, H., Ge, J., Gonzalez, J.E., Darrell, T., Chan, D.M.: Generate, but verify: Reducing hallucination in vision-language models with retrospective resampling. arXiv preprint arXiv:2504.13169 (2025)
59. Wu, W., Chang, T., Li, X., Yin, Q., Hu, Y.: Vision-language navigation: a survey and taxonomy. Neural Computing and Applications **36**(7), 3291–3316 (2024)
60. Yan, Y., Xu, R., Zhang, J., Li, P., Liang, X., Yin, J.: Instrugen: Automatic instruction generation for vision-and-language navigation via large multimodal models. arXiv preprint arXiv:2411.11394 (2024)
61. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., Cao, Y.: React: Synergizing reasoning and acting in language models. In: International Conference on Learning Representations (ICLR) (2023)
62. Zeng, S., Qi, D., Chang, X., Xiong, F., Xie, S., Wu, X., Liang, S., Xu, M., Wei, X.: Janusvln: Decoupling semantics and spatiality with dual implicit memory for vision-language navigation. arXiv preprint arXiv:2509.22548 (2025)
63. Zhang, Z., Zhu, W., Pan, H., Wang, X., Xu, R., Sun, X., Zheng, F.: Activevln: Towards active exploration via multi-turn rl in vision-and-language navigation. arXiv preprint arXiv:2509.12618 (2025)

64. Zheng, Y., Zhang, L., Sun, Y., Shen, Y., Zhao, S.: Canespeaker: An llm-assisted speaker for generating human-like navigation instructions. ACM Transactions on Multimedia Computing, Communications and Applications **22**(1), 1–26 (2026)
65. Zhou, A., Wang, Z., Levine, S., Finn, C.: Language feedback models for reinforcement learning. In: International Conference on Machine Learning (ICML) (2023)

Appendix

A Configuration of the Hindsight Speaker Agent

A.1 Video Parsing Strategy

We employ Φ-Nav on two representative VLN baselines: CMA [31], which follows a DAgger-based training scheme, and ETPNav [2], which adopts scheduled sampling. In addition to differences in their on-policy exploration strategies, the two baselines use different camera settings: CMA operates in a monocular setting, whereas ETPNav uses panoramic observations. We therefore tailor the visual input for our Hindsight Speaker Agent (HSA) to match the unique observation requirements of each baseline.

In the monocular setting, each timestep produces a visual observation consisting of a single 224×224 RGB image. The action space is also low-level, including `FORWARD`, `TURN LEFT`, `TURN RIGHT`, and `STOP`, which results in relatively long visual trajectory sequences. To improve efficiency, we subsample the trajectory by selecting one frame every two timesteps. Additionally, we resize the selected frames to 300×300 so that HSA can capture finer visual details.

In the panoramic setting used in ETPNav, each timestep produces a visual observation consisting of 12 RGB images of size 224×224 with a 90° field of view (FOV), where adjacent views overlap by 60°. To reduce redundancy, we select four non-overlapping views centered around the forward-facing direction and concatenate them horizontally to form a 224×896 image representing a 360° panoramic observation for each timestep. Since ETPNav abstracts continuous low-level actions into a topological waypoint-based action space, we use frames from all timesteps to construct the video sequence.

A.2 Hyperparameters

Next, we describe the hyperparameters used for HSA, which is based on the Qwen-VL [5] large vision-language model (LVLM). We set the temperature to 0.75 to encourage more diverse instruction generation while maintaining coherent outputs. The nucleus sampling parameter is set to top-$p = 0.9$ and top-$k = 50$ to allow controlled stochasticity while avoiding low-probability tokens. We enable sampling (`do_sample`=True) to encourage diverse yet coherent instruction outputs, and apply a repetition penalty of 1.1 to mitigate redundant phrasing. Finally, key-value caching (`use_cache`=True) is enabled to improve efficiency. All other hyperparameters follow the default settings of the respective baseline implementations.

A.3 Prompt Design

We present the prompt design used by HSA in Figure 5. Specifically, we adopt an in-context learning paradigm by providing a trajectory–instruction pair sampled from offline expert demonstrations. This example serves as a reference that guides the model to generate instructions consistent with the distribution of the training data. Consequently, the generated instructions are less likely to introduce noise or abrupt deviations during learning. In addition, we include explicit constraints in the prompt—covering grounding, style, flow, termination, and forbidden expressions—to encourage the pre-trained LVLM to produce high-quality and well-structured navigation instructions.

Expert-In-Context Hindsight Instruction Generation Prompt

System: You are an intelligent assistant that describes videos accurately.
Expert-In-Context Example:
– **Video:** {EXPERT_TRAJECTORY_VIDEO}
– **User Task:** Based *strictly* on the video sequence of the navigation path, generate a navigation instruction that best describes the path.
– **Constraints:**
  • **Grounding:** Refer to visible objects and landmarks in the video.
  • **Style:** Use natural and imperative language.
  • **Flow:** Combine steps into one fluid paragraph; do not list actions.
  • **Termination:** The last sentence must include the word that commands termination of navigation (e.g., 'stop', 'wait').
  • **Forbidden:** Do not use meta-words (e.g., 'image', 'frame', 'camera').
– **Generated Instruction:** {EXPERT_TRAJECTORY_INSTRUCTION}
Main Task:
– **Video:** {ON_POLICY_TRAJECTORY_VIDEO}
– **User Task:** <same user task as above>
– **Constraints:** <same constraints as above>
– **Generated Instruction:**

Fig. 5: Prompt used by the Hindsight Speaker Agent with an expert-in-context example to generate instructions from on-policy trajectories. Note that the <> marks are placeholders for actual texts.

B Distribution of TIAW Across Trajectories

In this section, we analyze the distribution of the Trajectory–Instruction Alignment Weight (TIAW) across trajectories. Specifically, we sample 5,000 distinct on-policy trajectories and compute TIAW across three different random seeds, as shown in Figure 6. Here, the x-axis represents the TIAW values discretized into 500 bins, while the y-axis indicates the corresponding frequency. We report the mean, standard deviation, minimum, and maximum values for each distribution. The results show that the weights are largely concentrated around 0.3, with most values ranging approximately from 0.1 to 0.6. This observation underscores Φ-Nav's ability to adaptively modulate the contribution of hindsight learning signals, assigning greater weight to semantically well-grounded trajectory–instruction pairs while down-weighting less aligned ones, thereby providing more informative and precise supervision for policy training.

C Computational Analysis

We evaluate the computational efficiency of Φ-Nav in the DAgger-based setting by measuring the time required to collect trajectory–instruction pairs for supervision. Specifically, we report two metrics: *trajectory-instruction per second* (TPS), which quantifies

Method	TPS↑	APS↑
CMA-D [31]	0.66	32.22
w/ Φ-Nav	**0.71**	**34.64**

Table 7: Computational Analysis

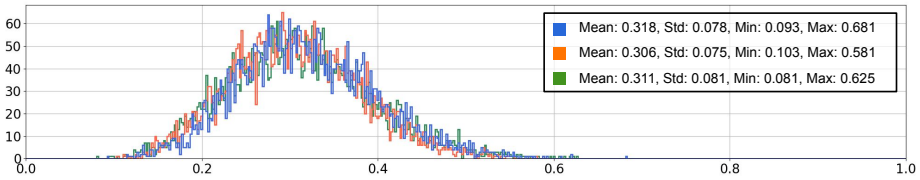

Fig. 6: Distribution of Trajectory–Instruction Alignment Weights.

the average number of on-policy trajectories paired with language instructions generated per second, and *action-instruction per second* (APS), which measures the average number of action–instruction pairs generated for those trajectories per second. APS can be interpreted as a step-wise version of TPS, providing a finer-grained view of computational throughput. As reported in Table 7, Φ-Nav enables the policy to collect a larger amount of supervision pairs within a given time budget of total 600 seconds. We acknowledge the inevitable increase in wall-clock training time due to the additional forward pass of the LVLM for instruction generation. However, this overhead is relatively minor when considering the substantial increase in the number of semantically grounded supervision obtained for policy learning.

D Additional Discussions

In this section, we highlight several notable questions and insights that emerged during the research process.

Question 1: How is Φ-Nav different from Speaker-Follower models?
Answer: While Φ-Nav naturally augments the set of training episodes, the contribution differs fundamentally from prior Speaker-Follower (SF)-based augmentation methods. First, the primary objective of SF is to increase the diversity of offline trajectory-instruction pairs, whereas Φ-Nav aims to address the semantic mismatch that arises during on-policy exploration. Furthermore, while SF generates instructions for static offline trajectories, Φ-Nav instead generates hindsight instructions conditioned on the agent's own exploratory trajectories. Consequently, Φ-Nav introduces challenges beyond conventional offline augmentation, including dynamically relabeling sub-optimal on-policy trajectories and preserving semantic consistency in hindsight supervision. Our proposed Expert-in-context Learning and TIAW, respectfully, are specifically designed to address these challenges, which do not arise in SF.

Question 2: Can Φ-Nav be applied in other embodied navigation tasks?
Answer: We believe that other embodied navigation tasks, such as object-goal navigation [9] or vision-based navigation [48], could potentially benefit from the idea of learning from hindsight experiences. However, VLN presents a particularly challenging setting due to the need to ground dense, long-horizon language instructions within complex visual environments. Φ-Nav is primarily designed for such scenarios by leveraging hindsight instruction generation to bridge the semantic supervision gap in on-policy training. Therefore, while the underlying principle of hindsight-based learning may extend to general embodied navigation tasks, Φ-Nav specifically addresses the unique challenges of semantic grounding and instruction following in language-driven navigation, which more closely resembles real-world human–robot interaction.

Question 3: Does Φ-Nav improve error recovery behavior?
Answer: Error recovery is a critical capability for embodied navigation agents, particularly in unseen environments where deviations from the optimal path frequently occur. While Φ-Nav does not explicitly introduce a dedicated mechanism for error recovery, it indirectly supports this behavior through improved semantic supervision during training. Specifically, Φ-Nav increases the density of semantically grounded trajectory–instruction pairs by generating hindsight instructions for on-policy exploratory trajectories. This enriched supervision expands the agent's semantic state coverage, enabling the policy to better associate environmental states with language-guided navigation behaviors. As a result, the agent becomes more capable of recognizing and correcting deviations during navigation, which contributes to improved performance in unseen environments and yields a positive effect on error recovery.

Question 4: What new research directions does Φ-Nav suggest?
Answer: Φ-Nav opens a pathway for embodied agents to self-analyze and reason over their own experiences. Considering both the contributions and limitations, we identify the following topics as promising directions for future research:

1. **Reducing Reliance on Offline Expert Demonstrations.**
   As demonstrated in Section 4.4 of the main manuscript, Φ-Nav achieves competitive performance compared to the baseline while using approximately 10% less offline expert demonstration data. This finding suggests a promising step toward mitigating the data-hungry nature of VLN training. Further efforts to reduce reliance on expert demonstrations could benefit not only VLN but also instruction-following embodied policy learning more broadly.

2. **Improving Trajectory-Instruction Alignment Scoring.**
   Just as generating high-quality instructions from trajectory videos is important, developing reliable numerical measures to assess trajectory–instruction alignment is equally critical. In this work, we employ a lightweight scoring mechanism inspired by EMScore [49]. However, developing more robust alignment metrics could further improve the effectiveness of hindsight supervision. Future work may explore leveraging 3D vision–language foundation models for better spatial grounding, as well as sequence-level alignment between visual trajectories and linguistic structures.

3. **Enabling Intermediate Hindsight Reasoning.**
   Φ-Nav enables embodied agents to retrospectively analyze their exploratory trajectories once navigation is completed. While this effectively bridges the semantic supervision gap in on-policy training, the training process could further benefit from intermediate hindsight reasoning during navigation, rather than only after termination. Such intermediate reflection may enable finer-grained path understanding across trajectories of varying lengths. However, introducing intermediate reasoning also raises challenges, including increased latency and the need to maintain spatial and temporal consistency throughout the trajectory, making this a challenging yet promising direction for future research.