The Fortieth AAAI Conference on Artificial Intelligence (AAAI-26)

SeqWalker: Sequential-Horizon Vision-and-Language Navigation with Hierarchical Planning

Zebin Han[1,2,3], **Xudong Wang**[2,4*], **Baichen Liu**[2], **Qi Lyu**[2,4], **Zhenduo Shang**[2,4], **Jiahua Dong**[5], **Lianqing Liu**[2], **Zhi Han**[2]
[1]North University of China
[2]State Key Laboratory of Robotics and Intelligent Systems, Shenyang Institute of Automation, Chinese Academy of Sciences
[3]Southeast University
[4]University of Chinese Academy of Sciences
[5]Mohamed bin Zayed University of Artificial Intelligence
{zebinhan04,dongjiahua1995}@gmail.com, {wangxudong,liubaichen,lvqi,shangzhenduo,lqliu,hanzhi}@sia.cn

Abstract

Sequential-Horizon Vision-and-Language Navigation (SH-VLN) presents a challenging scenario where agents should sequentially execute multi-task navigation guided by complex, long-horizon language instructions. Current vision-and-language navigation models exhibit significant performance degradation with such multi-task instructions, as information overload impairs the agent's ability to attend to observationally relevant details. To address this problem, we propose SeqWalker, a navigation model built on a hierarchical planning framework. Our SeqWalker features: i) A High-Level Planner that dynamically selects global instructions into contextually relevant sub-instructions based on the agent's current visual observations, thus reducing cognitive load; ii) A Low-Level Planner incorporating an Exploration-Verification strategy that leverages the inherent logical structure of instructions for trajectory error correction. To evaluate SH-VLN performance, we also extend the IVLN dataset and establish a new benchmark. Extensive experiments are performed to demonstrate the superiority of the proposed SeqWalker.

Code — https://seqwalker.github.io/seqwalker/

Introduction

Vision-and-Language Navigation (VLN) has emerged as a critical component for embodied AI, requiring intelligent agents to interpret natural language instructions and navigate through complex environments (Wei et al. 2025; Hong et al. 2024; Chen et al. 2024). This capability holds significant promise for diverse applications such as autonomous robotics (Wang et al. 2024d, 2023a), home robots (Gao et al. 2025), and virtual reality (Moshayedi et al. 2023). While recent advances (Bar et al. 2025; Zhao et al. 2024) have demonstrated remarkable progress in single-trajectory navigation scenarios, the more practical challenge of sequential multi-task navigation remains largely unaddressed.

In practical applications, agents are often required to execute multiple sub-tasks sequentially within a single deployment session (Song et al. 2024; Khanal et al. 2024). For example, on an usual morning, a student asks an agent to help him get up and go to school. This requires the agent to complete a series of tasks in sequence: i) take the clothes from the wardrobe, ii) take me to the bathroom for washing up, iii) prepare breakfast, iv) put the umbrella in my schoolbag, as illustrated in Fig.1. The navigation agent is required to complete the user's tasks sequentially and reach the destination. Different from the traditional VLN task, where the user describes one navigation trajectory with one instruction, the user only needs to provide the agent with one sequential-horizon instruction to enable the agent to complete multiple navigation tasks. This is more time-effective for the user, but it also poses more challenges. The sequential-horizon navigation means *larger navigation scenes* with *more complex user instructions*. Current VLN methods (Song et al. 2025; Yadav et al. 2025) are dedicated to persistent and efficient navigation in large scenes. Some methods (Zhang et al. 2024b; Zheng et al. 2024) store the historical state of previous processes and feed it into the agent's current state. Iterative Vision-and-Language Navigation (IVLN) (Krantz et al. 2023) proposes a map-based navigation method that maps the scene to maps during navigation. These advancements improve the navigation performance in persistent scenes and bring VLN research closer to large-scale scene applications. On the other hand, as the navigation scene expands, the complexity of the navigation task increases further, which means that the user describes multiple navigation tasks with one instruction, and the user inevitably describes multi-task trajectories with more complex and longer sequential-horizon instructions. However, the performance of existing VLN models is severely degraded with sequential-horizon instructions, as the excessive information hinders the agent from focusing on details relevant to its current observations.

To enable sequential multi-task navigation in large scenes, we introduce a new VLN task called *<u>S</u>equential-<u>H</u>orizon <u>V</u>ision-and-<u>L</u>anguage <u>N</u>avigation* (SH-VLN). This task emphasizes navigating long distances in large scenes using sequential-horizon instructions, which presents three main challenges: ***i) Scene Persistence.*** In real-world applications,

---

[*] Corresponding Author

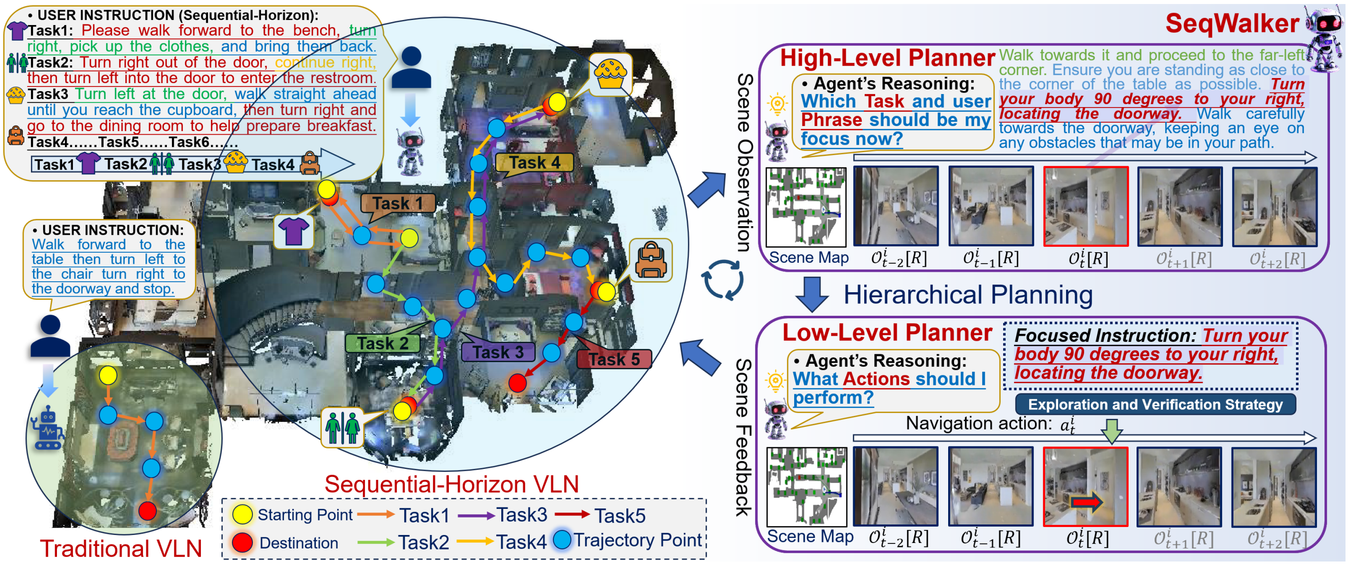

Figure 1: Illustration of the proposed *Sequential-Horizon Vision-and-Language Navigation* (SH-VLN) task and our SeqWalker model. Different from traditional VLN, SH-VLN requires navigation agents to follow a sequential multi-task trajectory navigation, while users tend to provide complex long instructions, posing greater challenges. Our SeqWalker adopts a *Hierarchical Planning* strategy, where the *High-Level Planner* selects sub-tasks and instruction phrases based on the agent's observations, and the *Low-Level Planner* provides actions for robust navigation using a proposed Exploration and Verification strategy.

a deployed navigation agent often operates within the same environment over extended periods (Jiang et al. 2025; Bai et al. 2025). To ensure efficient operation, the agent should be able to save scene information rather than treat each episode as a new exploration. ***ii) Complex Long Instructions.*** Different from a single-period navigation task, describing multi-task navigation in large scenes inevitably requires more complex and longer instructions to capture navigation trajectories. However, long and complex instructions interfere with the traditional navigation agents' understanding of instructions and make it difficult to match them to the current observation, leading to navigation failures. ***iii) Multi-Trajectory Robustness.*** Sequential-horizon navigation often means longer navigation trajectories. Sequential navigation in long trajectories is more likely to make errors, and once an error occurs, the trajectory is likely to worsen with it, so there are high demands on the robustness of navigation. An intuitive way to solve the long instruction problem is to directly leverage the powerful long sequence reasoning capability of large language models (LLMs). However, directly applying LLMs is ineffective. Although LLMs (OpenAI 2023; Zhang et al. 2025b) with several hundred billion parameters handle long sequences well, and models with smaller parameters that can be embedded in robots have shortcomings for long sequences. Additionally, directly applying LLM cannot guide accurate actions due to the lack of specific knowledge about the navigation scene.

To solve the SH-VLN task, we propose a SeqWalker, a novel VLN model built on a hierarchical planning framework. SeqWalker is capable of hierarchically understanding the user sequential long instructions step-by-step, while explicitly saving the scene maps to achieve efficient sequential-horizon navigation in large scenes, as shown in the Fig.1. Specifically, in *High-Level Planner*, SeqWalker performs local segmentation on global long instructions using pre-trained CLIP models to obtain the most relevant segmented instruction with the agent's current scenes. Compared to global long instructions, the segmentation instructions not only reduce the difficulty of focusing information but also enable corrections for navigation trajectory errors. In *Low-Level Planner*, we develop an Exploration and Verification strategy to achieve forward progress toward destinations or to leverage the inherent logical order of instructions to dynamically correct the current navigation trajectories, which further improves the robustness of long navigation. To evaluate the proposed SH-VLN task, we extend the existing IVLN dataset (Krantz et al. 2023) and propose a new benchmark. Extensive experiments are performed to demonstrate the effectiveness and superiority of our proposed SeqWalker.

The main contributions of this work are as follows:

- We introduce Sequential-Horizon Vision-and-Language Navigation (SH-VLN), a new VLN task where agents follow sequential-horizon instructions to complete a sequential multi-task navigation trajectory. We also provide a new benchmark for evaluating the SH-VLN task.
- We propose a SeqWalker model, which is built on a hierarchical planning framework. Our SeqWalker performs local segmentation on long instructions to understand instructions step-by-step while explicitly saving scene maps for sequential-horizon navigation in large scenes.
- We develop an Exploration and Verification strategy. Based on this, SeqWalker navigation in two modes, i.e., forward to destination and trajectory correction, further improving sequential-horizon navigation robustness.
- Extensive experiments show that SeqWalker achieves

state-of-the-art performance on both traditional IVLN datasets and the proposed sequential-horizon datasets.

Related Works

Vision-and-Language Navigation: The task of VLN (Liu et al. 2025; Tan et al. 2025; Chen et al. 2022; Zhang et al. 2025c) is to enable an agent to understand user instructions and navigate in complex environments independently. VLN tasks are typically divided into two types based on the environment: discrete (Gai et al. 2025; Wang et al. 2023b, 2024b) and continuous (Yu et al. 2025; Chen et al. 2025; Zhang et al. 2025a). In discrete environments, the space is modeled as a graph of connected nodes. The agent moves between these nodes following the given instructions, aiming to reach a target node within a set number of steps. In continuous environments, the agent can move freely using low-level actions such as moving forward or turning, offering more flexibility and a closer resemblance to real-world scenes. Since continuous environments align better with real applications, we follow the continuous VLN setting.

Navigation in Persistent Scenes: In real-world applications, navigation agents often operate in the same environment for extended periods (Bai et al. 2025; Zhang, Guo, and Kordjamshidi 2024; Zheng et al. 2025; Wang et al. 2024a). Meanwhile, advancements in computing power lead to richer scene datasets (Zhang et al. 2024a; Chang et al. 2017; Ramakrishnan et al. 2021) and improved simulator platforms (Kolve et al. 2017; Savva et al. 2019). These developments draw attention to the challenge of navigation in persistent scenes. Some studies (Wang et al. 2021; Zhang et al. 2024c) address persistent navigation by embedding information from past navigation episodes. However, merely storing scene information as hidden embeddings limits effectiveness. To improve long-term navigation, Iterative VLN (IVLN) task (Krantz et al. 2023) is introduced. IVLN uses a map-based method where it creates semantic maps of the scene during navigation and replaces the original image inputs with these maps. Although these advances bring VLN research closer to large-scene applications, the performance of curren VLN models drops significantly when handling multi-task instructions. We propose SH-VLN to tackle the challenges posed by sequential long-horizon instructions.

Problem Formulation

In VLN-CE task, an agent navigates a continuous scene Γ by following user instruction $\mathcal{I}$. In IVLN setting, a large scene Γ contains multiple trajectories $\mathcal{T} = \{T_0, T_1, \ldots, T_m\}$, and the agent persistently operates within Γ. In our SH-VLN task, instruction $\mathcal{I}$ is further extended to describe sequential-horizon trajectories $\mathcal{I} \Leftrightarrow \{T_0, T_1, \ldots, T_n\}$. Specifically, at each episode i, the user provides instruction $\mathcal{I}^i$ describing a sequential multi-task trajectory $\{T_0^i, T_1^i, \ldots, T_n^i\}$ in Γ, where $\mathcal{I}^i$ consists of phrases $\{S_0^i, S_1^i, \ldots, S_n^i\}$. At each time step t, the agent predicts the next navigation mode Exploration $\mathcal{E}_t^i$ or Verification $\mathcal{V}_t^i$ based on current observation $\mathcal{O}_t^i$. Mode $\mathcal{E}_t^i$ aims to move toward the destination, while $\mathcal{V}_t^i$ corrects trajectory errors. After multiple iterations, the agent reaches destination $\mathcal{G}^i$ with a sequential-horizon trajectory. Different from the previous LH-VLN task (Song et al. 2024), SH-VLN builds upon IVLN to evaluate efficient agent operation in persistent scenes. Beyond observation $\mathcal{O}^i$ and instruction $\mathcal{I}^i$, the agent also stores and leverages scene maps $\mathcal{M}^i$ from previous trajectories $\{T_0^i, T_1^i, \ldots, T_n^i\}$ in Γ for efficient navigation. Following (Krantz et al. 2020, 2023), agent actions within $\mathcal{E}_t^i$ are defined as $\mathcal{A}$ = {FORWARD (0.25m), TURN LEFT (15°), TURN RIGHT (15°), STOP}.

Proposed Methods

The pipeline of the proposed SeqWalker is illustrated in Fig.2, and the overall three navigation steps are as follows:

Step I. High-Level Perception Planner. SeqWalker agent matches the RGB $\mathcal{R}_t^i$ with the instruction $\mathcal{I}^i$, and selects out the phrase S_j^i most suitable for the agent's current state and observation to efficiently understand instruction.

Step II. Navigation Scene Mapping. Following IVLN-CE, to achieve agent's long-term planning in persistent large scenes, we use a <u>S</u>cene <u>M</u>apping <u>M</u>odule (SMM) to create and save scene maps, including semantic map $\mathcal{M}_t^i[sem]$ and occupancy $\mathcal{M}_t^i[ocu]$ map. And these saved scene maps are cropped for partial crops $\mathcal{M}c_t^i[sem]$ and $\mathcal{M}c_t^i[ocu]$ based on the agent current pose, to achieve efficient encoding with a Map-Encoder $Encoder_{map}$ for map embeddings $\mathcal{Z}_t^m$:

$$\mathcal{Z}_t^m = Encoder_{map}(\mathcal{M}c_t^i[sem], \mathcal{M}c_t^i[ocu]), \tag{1}$$

where $Encoder_{map}(\cdot)$ consists of four CBRA blocks, each CBRA includes a convolution, a normalization, a ReLU activation, and an averaging pool, with a spatial attention for focused feature extraction, i.e., $Encoder_{map} = [Attn_{cot}(CBRA(\cdot)) \times 4]$. More details of the proposed Map-Encoder are available in our Supplementary Material.

Step III. Low-Level Action Planner: We propose an Exploration and Verification (EaV) strategy to further improve the robustness of sequential-horizon navigation. The EaV offers two distinct navigation modes: exploration mode proactively guides agents toward destination, while verification mode validate and corrects trajectory errors in time.

High-Level Perception Planner

SeqWalker uses a High-Level Perception Planner to understand user instructions. In the SH-VLN task, it is difficult for the navigation agent to logically understand and process complex long instructions, thus, we propose an <u>I</u>nstruction <u>S</u>egmentation <u>M</u>odule (ISM) to segment global instructions to reduce the difficulty in understanding redundant instructions. The ISM decouples the sequential-horizon instruction into a series of sub-phrases, and the agent can focus on each sub-phrase sequentially for more accurate navigation.

Different from prior VLN models that encode instructions globally, we encode local instructions for a more accurate implementation. Specifically, for each phrase $\mathcal{S}_k^i$ in instruction $\mathcal{I}^i$ is encoded $\Psi_T(S_k^i)$ by Text-Encoder (CLIP (Radford et al. 2021)), and the observed RGB image $\mathcal{R}_t^i$ is encoded $\Psi_V(\mathcal{R}_t^i)$ by Vision-Encoder (CLIP). Then, the similarity θ_t^k between $\Psi_T(S_k^i)$ and $\Psi_V(\mathcal{R}_t^i)$ is calculated, and the probability of each phrase $\mathcal{P}(\mathcal{S}_k^i)$ matching the current state is:

$$\mathcal{P}(\mathcal{S}_k^i) = Softmax(\theta_t^k), \tag{2}$$

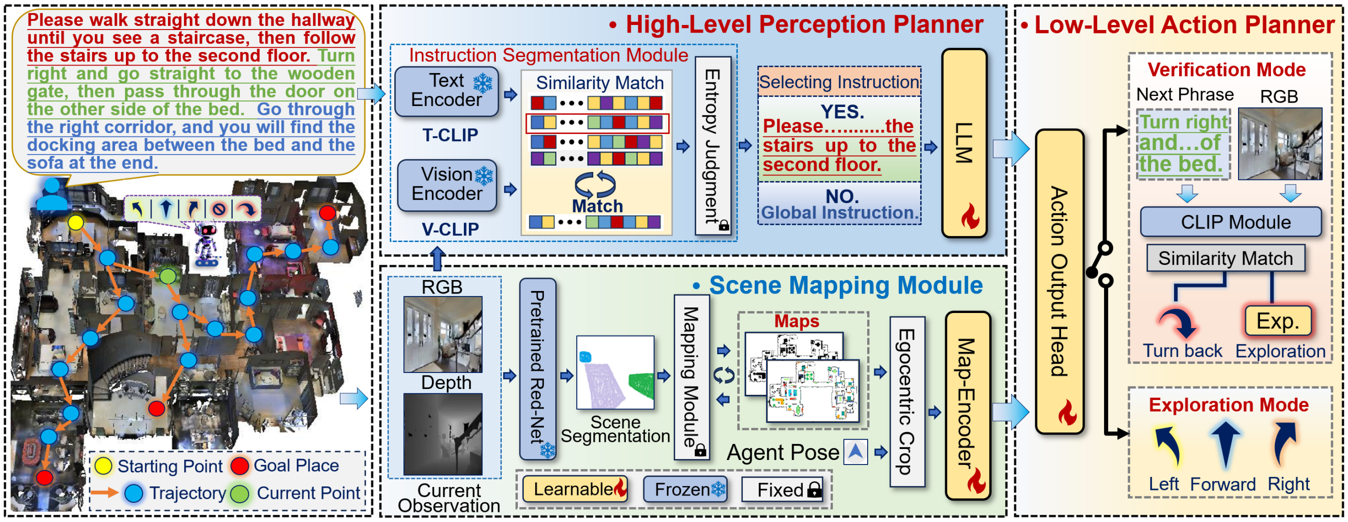

Figure 2: Illustration of the proposed SeqWalker pipeline. At each time step t, the SeqWalker agent receives RGB and depth observations to follow user instructions. SeqWalker has a hierarchical planning framework. It comprises a *High-Level Perception Planner*, which segments sequential-horizon instructions to obtain the most relevant segmented instruction with the current states. And it also comprises a *Low-Level Action Planner*, which employs an exploration-verification strategy to achieve progress toward destinations or leverage the inherent logical order of instructions to correct navigation trajectories dynamically.

$$\theta_t^k = Sim(\Psi_T(\mathcal{S}_k^i), \Psi_V(\mathcal{R}_t^i)), \tag{3}$$

where $Sim(\cdot)$ denotes the cosine similarity of the two inputs. However, simply selecting the phrase with the maximum $\mathcal{P}(\mathcal{S}_k^i)$ may cause an incorrect similar selection. For more robust selection, we further calculate the entropy value Φ_t^i for $\mathcal{P}(\mathcal{S}_k^i)$ to select the most suitable phrase $\mathcal{S}_{k^*}^i$:

$$\Phi_t^i = \mathcal{H}(\mathcal{P}(\mathcal{S}_0^i), \mathcal{P}(\mathcal{S}_1^i), \ldots, \mathcal{P}(\mathcal{S}_n^i)), \tag{4}$$

where $\mathcal{H}(\cdot)$ denotes entropy calculation, and when Φ_t^i is less than threshold Φ_λ, we take the phrase $\mathcal{S}_{k^*}^n$ with the maximum probability as the current focused local instruction:

$$k^* = \arg\max_k \left(\mathcal{P}(\mathcal{S}_k^i)\right), \tag{5}$$

otherwise, we take the global instruction $\mathcal{I}^i$ as the current navigation instruction to avoid incorrect instruction ignoring. And for richer semantic reasoning navigation, we use a lightweight LLM, i.e., Qwen-0.5b (Bai et al. 2023), as an Instruction-Encoder to obtain instruction embeddings $\mathcal{Z}_t^s$:

$$Z_t^s = LLM(\alpha \cdot \mathcal{S}_{k^*}^i + (1-\alpha) \cdot \mathcal{I}^i), \tag{6}$$

where $\alpha = 1$ if $\Phi_t^i < \Phi_\lambda$; otherwise, $\alpha = 0$. The LLM performs next token prediction for instruction encoding, and we take the embedding of the last hidden layer as the instruction embeddings $\mathcal{Z}_t^s$. This adaptive instruction encoding method facilitates agents in understanding complex instructions.

Low-Level Action Planner

SeqWalker uses a Low-Level Action Planner to provide specific navigation actions. In the SH-VLN task, the navigation multi-trajectories are long and sequentially strict. The agent may take an incorrect trajectory due to a biased understanding of the semantics of some instructions, and the current error often causes subsequent navigation to fail irreversibly. To improve the robustness of the navigation trajectory, we further propose an Exploration and Verification strategy as the Low-Level Action Planner for navigation.

Exploration Navigation Mode: The Exploration Navigation Mode moves the agent toward the destination. The agent integrates the above embeddings in Exploration Mode for action prediction. Specifically, the obtained embeddings $\mathcal{Z}_t^m$, $\mathcal{Z}_t^s$ are jointly loaded into an Action Output Head (AOH) and are combined with the previous hidden state h_{t-1}^i and action a_{t-1}^i to predict the next action a_t^i:

$$a_t^i, h_t^a = AOH(\mathcal{Z}_t^m, \mathcal{Z}_t^s, h_{t-1}^i, a_{t-1}^i), \tag{7}$$

where $AOH(\cdot)$ consists of two Gated Recurrent Units (GRU) with cross attention blocks, referring (Krantz et al. 2023). The $AOH(\cdot)$ outputs the probability of each action $\mathcal{A}$, and we take the action corresponding to the maximum probability as a_t^i. a_t^i moves agent toward destination.

Verification Navigation Mode: The Verification Navigation Mode corrects the errors on the current navigation trajectory. Based on Exploration Mode, the agent can actively search for destinations in navigation scenes. However, the agent unavoidably misjudges the direction of forks due to sequential-horizon navigation trajectories. We further propose a Verification Mode that utilizes the internal logical order of the instructions to correct navigation trajectories. Specifically, the order of the agent navigation process should match the sequence of the phrases. When performing the previous action a_{t-1}^i and switching phrases, the new phrase $\mathcal{S}_{k^*}^n$ should be the next phrase of the last selected phrase $\mathcal{S}_{k_l^*}^n$, i.e., $\mathcal{S}_{k^*}^n = \mathcal{S}_{k_l^*+1}^n$. And during navigation, the agent's current observation $\mathcal{O}_t^i[R]$ should have a high similarity to the selected phrase $\mathcal{S}_{k_l^*+1}^n$. Based on this logic, we conduct two terms verification. Firstly, the agent determines whether the currently selected phrase is $\mathcal{S}_{k_l^*+1}^n$ (Term-I). Secondly, the

Algorithm 1: Exploration and Verification Strategy

```
Input: Current observation $\mathcal{O}_t^i$, current action $a_t^i$, user's instruction $\mathcal{I}^i$, ver-threshold value $\delta_0$ and $\alpha$ from Eq. (6).
Obtain the current RGB observation: $\mathcal{O}_t^i[R]$;
Recall the last selected phrase: $S_{k_l^*}^i$, and compute the currently selected phrase or global instruction: $S_{k^*}^i$;
if $\alpha = 0$ or $k^* = (k_l^* + 1)$ then
    Predict the next action $a_t^i$ with Eq. (7).
else
    text_embedding = $\Psi_T(S_{k_l^*+1}^i)$;
    rgb_embedding = $\Psi_V(\mathcal{O}_t^i[R])$;
    $\delta_t$ = $sim$(rgb_embedding, text_embedding).
    if $\delta_t < \delta_0$ then
        $a_t^i$ = {"TURN_BACK_LAST_STEP"};
        $mode$ = "Ver": mandatory setting of the next action to take the action corresponding to the second highest probability.
    else
        Predict the next action $a_t^i$ with Eq. (7).
    end if
end if
Perform $a_t^i$ to obtain new observation state $\mathcal{O}_t^i$.
```

agent compares the similarity δ_t between $\mathcal{S}_{k^*+1}^n$ and $\mathcal{O}_t^i[R]$ with a ver-threshold value δ_0 (Term-II). Specifically, when the phrase selected $\mathcal{S}_{k^*}^n$ is not $\mathcal{S}_{k_l^*+1}^n$ and $\delta_t < \delta_0$, the agent returns to the previous position where the a_{t-1}^i performed, where δ_0 is a learnable parameter determined during navigation training for better adaptive navigation. And the specific steps of EaV strategy are summarized in Algorithm 1.

Learning Sequential-Horizon Navigation

Sequential-Horizon Dataset Transformation. To evaluate the proposed SH-VLN task, we extend the IVLN IR2R-CE dataset (Krantz et al. 2023) and propose a new benchmark. Two key aspects are considered in constructing this benchmark: how to make the single-stage navigation trajectories be stitched together into sequential-horizon trajectories, and how to generate long instructions with more discriminating details that closely align with navigation scenes.

a). Construction Sequential-Horizon Trajectories. Different from the one-to-one mapping between instruction and trajectory in IR2R-CE, the sequential-horizon navigation task involves a single instruction spanning multiple trajectories. To construct such trajectories, we select and connect pairs from the IR2R-CE dataset whose end- and start-points align. Corresponding instructions are concatenated to ensure semantic coherence using LLM, i.e., LLaMa-13B (Touvron et al. 2023). The concatenation prompt template is:

> ***User: Please help logically connect two navigation instructions into one, ensuring semantic coherence, with the end of the first serving as the start of the second.*** **<INS1><INS2>**

b). Enrichment Long Instructions. In addition to generating sequential-horizon trajectories, we also enrich the instructions to address their lack of sufficient discriminative granularity to distinguish semantically similar subtasks in multi-trajectory navigation. Specifically, we extend the IVLN instructions based on a large multimodal language model, i.e., LLaVA-OneVision (Li et al. 2024), that can reason multiview images with language prompts. Specifically, during instruction enrichment, an agent is forced to follow the ground-truth trajectories for step-by-step navigation. In the navigation trajectory corresponding to each phrase S_k^i, we collect the observed RGB images to construct multiple images set $\{R_0^k, R_1^k, \ldots, R_n^k\}$ of a first view corresponding to a navigation trajectory. Then, we feed the phrase S_k^i and multiple images as inputs into LLaVA to implement instruction enrichment. The used enrichment prompt template is:

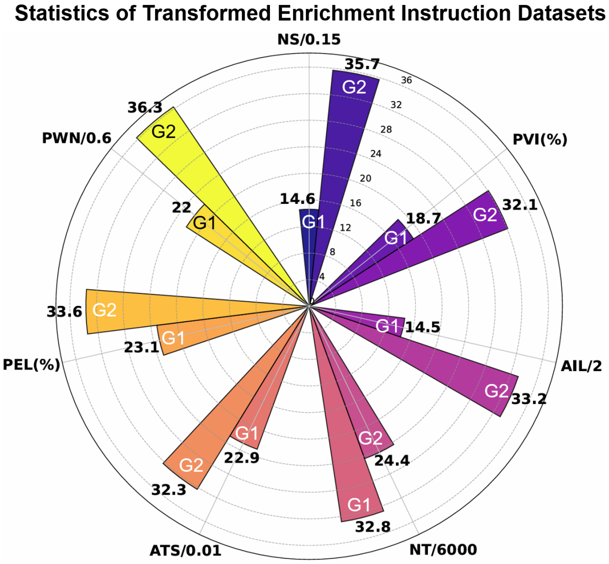

Figure 3: Statistics of the transformed datasets. The statistics graph contains seven groups of comparison metrics: Number of Sentences, Percentage of Verbs in Instruction, Average Instruction Length, Number of Trajectories, Percentage of Explicit Locations, Average Trajectories Scene, Phrases Word Number, **G1** represents unchanged instructions, and **G2** represents our transformed enrichment IR2R-CE dataset. Please note that for better visualization, we mark some metrics as "A/x", i.e., and the true value is $A \times x$.

> ***User: <IMAGES> Please look closely at these multiple images of the first view corresponding to a navigation trajectory, and help me enrich the discriminating details of instruction without changing its logic. The original instruction is:*** **<INS> Once you enter the bed room, go straight to the left-hand door near the table.**

where the bold is the fixed enrichment prompt and the **<INS>** part is the user instruction phrase. The **<IMAGES>** represents the tokens containing the multiple images set $\{R_0^k, R_1^k, \ldots, R_n^k\}$ of a first view corresponding to a navigation trajectory. We save the enriched instruction output provided by LLaVA and replace phrases S_k^i. The process is repeated to perform the expansion for all instructions.

Based on the above two main aspects, we construct the Sequential-Horizon IR2R-CE (SH IR2R-CE), and a large-

	Val-Seen							Val-Unseen						
Comparisons	TL	OS↑	nDTW↑	SR↑	SPL↑	CPsubT↑	t-nDTW↑	TL	OS↑	nDTW↑	SR↑	SPL↑	CPsubT↑	t-nDTW↑
CMA	12.5$_{\pm 0.8}$	21$_{\pm 1}$	35$_{\pm 3}$	11$_{\pm 3}$	8$_{\pm 2}$	33$_{\pm 1}$	31$_{\pm 2}$	12.1$_{\pm 0.3}$	16$_{\pm 1}$	36$_{\pm 2}$	9$_{\pm 2}$	6$_{\pm 1}$	30$_{\pm 2}$	30$_{\pm 1}$
TourCMA	12.8$_{\pm 0.6}$	23$_{\pm 3}$	36$_{\pm 2}$	14$_{\pm 2}$	10$_{\pm 1}$	38$_{\pm 2}$	32$_{\pm 1}$	12.5$_{\pm 0.3}$	19$_{\pm 1}$	33$_{\pm 1}$	11$_{\pm 2}$	8$_{\pm 2}$	38$_{\pm 1}$	28$_{\pm 2}$
PoolCMA	12.2$_{\pm 0.7}$	19$_{\pm 2}$	33$_{\pm 2}$	11$_{\pm 3}$	9$_{\pm 2}$	36$_{\pm 1}$	29$_{\pm 2}$	12.3$_{\pm 0.6}$	18$_{\pm 1}$	31$_{\pm 1}$	13$_{\pm 1}$	10$_{\pm 2}$	34$_{\pm 2}$	27$_{\pm 2}$
Po.E.CMA	12.2$_{\pm 0.8}$	22$_{\pm 3}$	37$_{\pm 3}$	13$_{\pm 2}$	11$_{\pm 2}$	42$_{\pm 1}$	33$_{\pm 2}$	11.6$_{\pm 0.7}$	20$_{\pm 3}$	38$_{\pm 2}$	12$_{\pm 2}$	10$_{\pm 1}$	41$_{\pm 3}$	32$_{\pm 2}$
MAP-CMA	14.5$_{\pm 0.6}$	35$_{\pm 3}$	46$_{\pm 3}$	26$_{\pm 3}$	22$_{\pm 2}$	53$_{\pm 1}$	43$_{\pm 1}$	13.5$_{\pm 0.6}$	31$_{\pm 2}$	44$_{\pm 2}$	23$_{\pm 1}$	20$_{\pm 1}$	50$_{\pm 1}$	39$_{\pm 1}$
ETPNav	13.9$_{\pm 0.7}$	30$_{\pm 3}$	41$_{\pm 2}$	23$_{\pm 2}$	20$_{\pm 1}$	46$_{\pm 2}$	38$_{\pm 1}$	13.1$_{\pm 0.5}$	29$_{\pm 2}$	38$_{\pm 1}$	20$_{\pm 1}$	16$_{\pm 1}$	44$_{\pm 1}$	32$_{\pm 1}$
HNR	14.0$_{\pm 0.6}$	31$_{\pm 3}$	42$_{\pm 3}$	24$_{\pm 4}$	21$_{\pm 2}$	49$_{\pm 1}$	39$_{\pm 1}$	13.3$_{\pm 0.7}$	30$_{\pm 2}$	39$_{\pm 2}$	21$_{\pm 1}$	17$_{\pm 1}$	45$_{\pm 2}$	33$_{\pm 2}$
SeqWalker	17.8$_{\pm 1.2}$	**43**$_{\pm 2}$	**52**$_{\pm 2}$	**34**$_{\pm 1}$	**33**$_{\pm 2}$	**67**$_{\pm 2}$	**48**$_{\pm 1}$	17.3$_{\pm 1.0}$	**40**$_{\pm 2}$	**50**$_{\pm 1}$	**30**$_{\pm 2}$	**29**$_{\pm 1}$	**66**$_{\pm 3}$	**45**$_{\pm 1}$

Table 1: The test results for SeqWalker compared to SOTA methods on the ***SH IR2R-CE datasets***. TL is in meters, and OS, nDTW, SR, SPL, CPsubT, t-nDTW are reported as percentages. Results are presented as mean ± standard deviation.

	Val-Seen							Val-Unseen						
Comparisons	TL	NE↓	OS↑	nDTW↑	SR↑	SPL↑	t-nDTW↑	TL	NE↓	OS↑	nDTW↑	SR↑	SPL↑	t-nDTW↑
CMA	7.8$_{\pm 0.4}$	8.8$_{\pm 0.6}$	27$_{\pm 3}$	42$_{\pm 3}$	18$_{\pm 3}$	17$_{\pm 3}$	39$_{\pm 1}$	7.5$_{\pm 0.3}$	8.8$_{\pm 0.2}$	26$_{\pm 1}$	44$_{\pm 1}$	19$_{\pm 1}$	18$_{\pm 1}$	38$_{\pm 2}$
TourCMA	8.0$_{\pm 0.4}$	8.2$_{\pm 0.9}$	30$_{\pm 2}$	44$_{\pm 2}$	20$_{\pm 3}$	19$_{\pm 2}$	40$_{\pm 1}$	7.8$_{\pm 0.1}$	9.0$_{\pm 0.2}$	26$_{\pm 1}$	42$_{\pm 1}$	18$_{\pm 0}$	17$_{\pm 1}$	36$_{\pm 1}$
PoolCMA	7.2$_{\pm 0.5}$	9.1$_{\pm 0.4}$	24$_{\pm 4}$	41$_{\pm 2}$	17$_{\pm 4}$	16$_{\pm 2}$	37$_{\pm 2}$	7.3$_{\pm 0.2}$	9.0$_{\pm 0.3}$	23$_{\pm 1}$	42$_{\pm 1}$	16$_{\pm 1}$	15$_{\pm 0}$	36$_{\pm 2}$
Po.E.CMA	7.6$_{\pm 0.8}$	8.9$_{\pm 0.9}$	27$_{\pm 3}$	42$_{\pm 3}$	18$_{\pm 4}$	17$_{\pm 2}$	38$_{\pm 2}$	6.9$_{\pm 0.2}$	8.7$_{\pm 0.2}$	25$_{\pm 2}$	44$_{\pm 1}$	18$_{\pm 1}$	16$_{\pm 1}$	38$_{\pm 2}$
MAP-CMA	9.4	6.4	48	56	39	36	52	8.5	6.8	44	54	35	32	47
ETPNav	9.6$_{\pm 0.9}$	8.8$_{\pm 0.9}$	45$_{\pm 4}$	52$_{\pm 2}$	36$_{\pm 2}$	31$_{\pm 1}$	47$_{\pm 2}$	8.9$_{\pm 0.6}$	8.6$_{\pm 0.2}$	37$_{\pm 2}$	47$_{\pm 2}$	28$_{\pm 1}$	27$_{\pm 1}$	41$_{\pm 2}$
HNR	10.8$_{\pm 0.5}$	7.9$_{\pm 0.9}$	46$_{\pm 4}$	53$_{\pm 3}$	37$_{\pm 2}$	33$_{\pm 2}$	50$_{\pm 1}$	9.3$_{\pm 0.7}$	7.5$_{\pm 0.4}$	40$_{\pm 2}$	50$_{\pm 1}$	30$_{\pm 1}$	28$_{\pm 2}$	44$_{\pm 2}$
OVER-NAV	9.5$_{\pm 0.9}$	5.8$_{\pm 0.9}$	49$_{\pm 4}$	59$_{\pm 2}$	39$_{\pm 2}$	36$_{\pm 2}$	56$_{\pm 2}$	8.8$_{\pm 0.6}$	6.5$_{\pm 0.2}$	45$_{\pm 2}$	56$_{\pm 1}$	35$_{\pm 1}$	33$_{\pm 1}$	50$_{\pm 2}$
SeqWalker	12.3$_{\pm 0.8}$	**5.5**$_{\pm 0.7}$	**51**$_{\pm 3}$	**60**$_{\pm 2}$	**41**$_{\pm 2}$	**38**$_{\pm 2}$	**58**$_{\pm 2}$	11.4$_{\pm 0.5}$	**6.4**$_{\pm 0.2}$	**46**$_{\pm 2}$	**58**$_{\pm 2}$	**36**$_{\pm 2}$	**34**$_{\pm 2}$	**52**$_{\pm 2}$

Table 2: The test results for SeqWalker compared to SOTA methods on the ***IR2R-CE datasets*** (Krantz et al. 2023). TL and NE are in metres, and OS, nDTW, SR, SPL, and t-nDTW are reported as percentages. The results are reported as $\bar{x} \pm \sigma_{\bar{x}}$.

scale collection of single trajectories paired with corresponding instructions is used for training and testing. The statistics for the constructed dataset are presented in Fig.3

Experiments

Implementation

We use Habitat (Savva et al. 2019) as the simulation platform. Our SeqWalker is trained on the proposed SH IR2R-CE dataset. We use a lightweight LLM, i.e., Qwen-0.5b (Bai et al. 2023), as an Instruction-Encoder to obtain instruction embeddings, and CLIP ViT-B/32 as the image-text encoders for the high-level perception planner, and we set the threshold $\Phi_\lambda = 0.65$, the Term-I and Term-II for turning back are included in verification mode. We evaluate the performance following previous metrics (Anderson et al. 2018; Ilharco et al. 2019; Krantz et al. 2023): Navigation Error (NE), Oracle Success Rate (OS), normalized Dynamic Time Warping (nDTW), Success Rate (SR), Success weighted by Path Length (SPL), Tour normalized Dynamic Time Warping (t-nDTW), and Trajectory Length (TL). We also propose two metrics to evaluate sub-instruction task completion: $CP_{sub}T$, the ratio of completed sub-tasks before failure, $CP_{sub}T = NS/NA$, where NS and NA are the numbers of completed and total sub-tasks, respectively. $CP_{sub}I$, the proportion of correctly selected sub-instructions, defined as $CP_{sub}I = NC/NT$, where NC is the number of correct selections based on entropy and NT is the total steps.

Training Strategy. The proposed SeqWalker adopts a two-stage imitation training strategy, following (Krantz et al. 2020). The first stage of the training process uses teacher forcing on the proposed SH-IR2R-CE datasets. In the second stage, the model of the first stage is fine-tuned to achieve a better generalization performance in unseen scenes. We use the Progress Monitor auxiliary loss in both training steps. The modules in SeqWalker are partially trainable, as shown in Fig.2. We freeze the parameters of CLIP models to preserve their text-image matching and understanding capabilities. The LLM-based Instruction-Encoder, Maps-Encoder, and AOH are jointly trained during the training.

Comparison Experiments

How does SeqWalker perform with sequential-horizon instructions? This experiment is used to verify the performance of SeqWalker on the sequential-horizon navigation task. To ensure fairness, all methods are trained on the proposed SH IR2R-CE dataset. The SOTA comparison models include CMA (Krantz et al. 2023) series are Naive-CMA, PoolCMA, PoolEndCMA, etc. and HNR (Wang et al. 2024c), ETPNav (An et al. 2024). A summary of the evaluation results can be found in Table 1. SeqWalker achieves superior navigation performance with long instructions, improving the t-nDTW metric by 5% and 6% over the previous best model on the val-seen and val-unseen sets, respectively. Significant improvements are observed across multiple metrics, e.g., SR, t-nDTW, etc. For methods without local instruction awareness, $CP_{sub}I$ is 0, whereas SeqWalker achieves 74%, demonstrating effective sub-instruction comprehension during navigation. The results show that SeqWalker has superior performance in the SH-VLN task.

#	ISM	LLM-Size	TL	NE↓	OS↑	nDTW↑	SR↑	SPL↑	t-nDTW↑
1	✓	0.5B	17.3	**6.6**	**40**	**50**	**30**	**29**	**45**
2	✗	0.5B	17.5	7.9	33	44	25	21	38
3	✗	7.0B	17.2	7.3	36	46	27	23	41
4	✗	13B	16.8	7.0	37	48	27	24	42

Table 3: Ablation study on Val-Unseen split. Instruction encoding consistently improves with larger LLMs, while ISM brings notable gains for smaller and mid-sized ones.

#	Model	TL	NE↓	OS↑	nDTW↑	SR↑	SPL↑	t-nDTW↑
1	$Maximum$	17.2	7.3	36	47	27	25	42
2	$\Phi_\lambda = 0.85$	17.5	7.2	38	48	27	26	43
3	$\Phi_\lambda = 0.75$	16.8	6.9	39	48	29	28	44
4	$\Phi_\lambda = 0.65$	17.3	**6.6**	**40**	**50**	**30**	**29**	**45**
5	$\Phi_\lambda = 0.55$	17.4	7.0	37	47	28	26	43
6	$\Phi_\lambda = 0.45$	17.2	7.4	37	46	27	25	42

Table 4: Ablation study for **phrase selection method** and different **entropy threshold values** Φ_λ on the Val-Unseen.

How does SeqWalker perform with traditional single-task instructions? We also conduct experiments to verify the navigation performance on the traditional IVLN task. The SOTA comparison models also include the CMA family (Krantz et al. 2023) of Naive-CMA, PoolCMA, PoolEnd-CMA, MAP-CMA, and OVER-NAV (Zhao et al. 2024). A summary of the results can be found in Table 2. Leveraging the effectiveness of the hierarchical planning framework, SeqWalker also has superior performance on the IVLN task.

Ablation Studies

What are the advantages of our ISM over simply using LLM with a larger number of parameters? We study the effectiveness of ISM improving instruction encoding, and Table 3 summarizes the ablation results. Although instruction encoding improves with increasing LLM parameters, our ISM significantly enhances LLM performance with smaller models. Our ISM splits long instructions into subtasks, which significantly improves the reasoning of small-parameter LLM. This is crucial in real-world robotics applications, where real-time constraints and limited on-device computation hinder the deployment of large LLMs.

What is the best entropy threshold for segmentation? We first provide experiment results for simply selecting phrases with maximum similarity, denoted as $Maximum$. We also study the effect of different entropy thresholds Φ_λ on navigation performance; ablation results are summarized in Table 4. As Φ_λ decreases, navigation performance initially improves, peaking near $\Phi_\lambda = 0.65$, then decreases (see Table 4 "#2-6"). Higher thresholds allow more phrases to be used, improving navigation. Excessive Φ_λ harms prediction from global instructions and causes incorrect choices among similar phrases, reducing performance.

How do different segmentation styles perform? We study the effectiveness of instruction segmentation and the effect of different segmentation styles on navigation performance, and the ablation results are summarized in Table 5. Type-I indicates the instructions are not segmented. Type-II indicates that the instructions are segmented based on commas. Type-III indicates that the instructions are segmented using conjunctions such as "and" or "then". Type-IV indicates that the instructions are segmented based on period. The results demonstrate that instruction segmentation has a significant improvement on sequential-horizon instruction navigation (referring Table 5 "#1"). The different segmentation styles have a significant affects navigation performance. Type-IV achieves the best results, as its comma-based segmentation ensures each phrase includes at least one object, facilitating better alignment with visual observations.

#	Model	TL	NE↓	OS↑	nDTW↑	SR↑	SPL↑	t-nDTW↑
1	Type-I	16.4	8.2	32	45	23	21	39
2	Type-II	16.7	6.9	36	46	26	24	42
3	Type-III	16.7	7.1	36	45	25	24	41
4	Type-IV	17.3	**6.6**	**40**	**50**	**30**	**29**	**45**

Table 5: Ablation study of the proposed **ISM** on Val-Unseen.

#	Ver.	Term-I	Term-II	TL	NE↓	OS↑	nDTW↑	SR↑	SPL↑	t-nDTW↑
1	✗	✗	✗	12.9	7.4	35	44	25	23	40
2	✓	✓	✗	17.8	6.8	38	48	28	27	43
3	✓	✗	✓	17.1	7.1	38	47	27	25	42
4	✓	✓	✓	17.3	**6.6**	**40**	**50**	**30**	**29**	**45**

Table 6: The ablation study for the proposed **EaV strategy** on the proposed SH IR2R Val-Unseen Scenes.

How effective is the EaV strategy? We study the effect of the proposed EaV strategy, including Term-I and Term-II in Verification Mode. The ablation results are summarized in Table 6. This ablation study demonstrates that the proposed EaV strategy can significantly improve the success rate of navigation. Although the trajectory correction procedures introduce additional trajectory length, they contribute to notable improvements in navigation success rates and other related metrics, e.g., task completion and trajectory quality. When Term-I and Term-II are both included in EaV strategy, our SeqWalker achieves robust sequential-horizon navigation with the best performance (referring Table 6 "#4").

Conclusion

In this paper, we introduce a new VLN task called Sequential-Horizon Vision-and-Language Navigation. To solve the SH-NLV task, we propose SeqWalker, a novel navigation model built on a hierarchical planning framework. In High-Level Planner, SeqWalker performs local segmentation on global long instructions to obtain the most relevant segmented instruction with the agent's current scenes. In Low-Level Planner, we develop an Exploration and Verification strategy to achieve forward progress toward the destination or to leverage the inherent logical order of instructions to dynamically correct the error of current navigation trajectories. To evaluate the SH-VLN performance, we extend the IVLN dataset and establish a new benchmark. Extensive experiments validate SeqWalker's superior performance.

Acknowledgments

This work was supported by the National Key Research and Development Program of China under Grant 2024YFB4707700, the National Natural Science Foundation of China under Grant T2596045, T2596040 and U23A20343, CAS Project for Young Scientists in Basic Research, Grant YSBR-041, Liaoning Provincial "Selecting the Best Candidates by Opening Competition Mechanism" Science and Technology Program under Grant 2023JH1/10400045, Fundamental Research Project of SIA under Grant 2024JC3K01.

References

An, D.; Wang, H.; Wang, W.; Wang, Z.; Huang, Y.; He, K.; and Wang, L. 2024. ETPNav: Evolving Topological Planning for Vision-Language Navigation in Continuous Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–16.

Anderson, P.; Chang, A. X.; Chaplot, D. S.; Dosovitskiy, A.; Gupta, S.; Koltun, V.; Kosecka, J.; Malik, J.; Mottaghi, R.; Savva, M.; and Zamir, A. 2018. On Evaluation of Embodied Navigation Agents. *ArXiv*, abs/1807.06757.

Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; and et al., K. D. 2023. Qwen Technical Report. *arXiv preprint arXiv:2309.16609*.

Bai, Z.; Li, H.; Fu, B.; Xiong, C.; Wang, R.; and Chen, X. 2025. R2C: Mapping Room to Chessboard to Unlock LLM As Low-Level Action Planner. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 19456–19466.

Bar, A.; Zhou, G.; Tran, D.; Darrell, T.; and LeCun, Y. 2025. Navigation world models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 15791–15801.

Chang, A. X.; Dai, A.; Funkhouser, T. A.; Halber, M.; Nießner, M.; Savva, M.; Song, S.; Zeng, A.; and Zhang, Y. 2017. Matterport3D: Learning from RGB-D Data in Indoor Environments. *International Conference on 3D Vision*, 667–676.

Chen, B.; Kang, J.; Zhong, P.; Liang, Y.; Sheng, Y.; and Wang, J. 2024. Embodied Contrastive Learning with Geometric Consistency and Behavioral Awareness for Object Navigation. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24, 4776–4785. New York, NY, USA: Association for Computing Machinery. ISBN 9798400706868.

Chen, J.; Lin, B.; Liu, X.; Ma, L.; Liang, X.; and Wong, K.-Y. K. 2025. Affordances-oriented planning using foundation models for continuous vision-language navigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 23568–23576.

Chen, P.; Ji, D.; Lin, K.; Zeng, R.; Li, T.; Tan, M.; and Gan, C. 2022. Weakly-Supervised Multi-Granularity Map Learning for Vision-and-Language Navigation. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *Advances in Neural Information Processing Systems*, volume 35, 38149–38161. Curran Associates, Inc.

Gai, J.; Guo, Z.; Raj, A.; and Tang, L. 2025. Robust crop row detection using discrete Fourier transform (DFT) for vision-based in-field navigation. *Computers and Electronics in Agriculture*, 229: 109666.

Gao, Y.; Chang, Y.; Yang, T.; and Yu, Z. 2025. Consumer acceptance of social robots in domestic settings: A human-robot interaction perspective. *Journal of Retailing and Consumer Services*, 82: 104075.

Hong, H.; Wang, S.; Huang, Z.; Wu, Q.; and Liu, J. 2024. Navigating Beyond Instructions: Vision-and-Language Navigation in Obstructed Environments. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 7639–7648.

Ilharco, G.; Jain, V.; Ku, A.; Ie, E.; and Baldridge, J. 2019. General Evaluation for Instruction Conditioned Navigation using Dynamic Time Warping. In *NeurIPS Visually Grounded Interaction and Language Workshop*, volume abs/1907.05446.

Jiang, J.; Zhu, Y.; Wu, Z.; and Song, J. 2025. DualMap: Online Open-Vocabulary Semantic Mapping for Natural Language Navigation in Dynamic Changing Scenes. *arXiv preprint arXiv:2506.01950*.

Khanal, A.; Bui, H.-D.; Plaku, E.; and Stein, G. J. 2024. Learning-informed Long-Horizon Navigation under Uncertainty for Vehicles with Dynamics. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3069–3075.

Kolve, E.; Mottaghi, R.; Han, W.; VanderBilt, E.; Weihs, L.; Herrasti, A.; Deitke, M.; Ehsani, K.; Gordon, D.; Zhu, Y.; Kembhavi, A.; Gupta, A. K.; and Farhadi, A. 2017. AI2-THOR: An Interactive 3D Environment for Visual AI. *ArXiv*, abs/1712.05474.

Krantz, J.; Banerjee, S.; Zhu, W.; Corso, J.; Anderson, P.; Lee, S.; and Thomason, J. 2023. Iterative Vision-and-Language Navigation. In *Computer Vision and Pattern Recognition*, 14921–14930.

Krantz, J.; Wijmans, E.; Majumdar, A.; Batra, D.; and Lee, S. 2020. Beyond the Nav-Graph: Vision-and-Language Navigation in Continuous Environments. In *European Conference on Computer Vision*, 104–120. Cham: Springer International Publishing. ISBN 978-3-030-58604-1.

Li, B.; Zhang, Y.; Guo, D.; Zhang, R.; Li, F.; Zhang, H.; Zhang, K.; Li, Y.; Liu, Z.; and Li, C. 2024. LLaVA-OneVision: Easy Visual Task Transfer. *ArXiv*, abs/2408.03326.

Liu, Y.; Liu, L.; Zheng, Y.; Liu, Y.; Dang, F.; Li, N.; and Ma, K. 2025. Embodied navigation. *Science China Information Sciences*, 68: 1–39.

Moshayedi, A. J.; Reza, K. S.; Khan, A. S.; and Nawaz, A. 2023. Integrating virtual reality and robotic operation system (ROS) for AGV navigation. *EAI Endorsed Transactions on AI and Robotics*, 2.

OpenAI. 2023. GPT-4 Technical Report. *CoRR*, abs/2303.08774.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.;

et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, 8748–8763. PMLR.

Ramakrishnan, S. K.; Gokaslan, A.; Wijmans, E.; Maksymets, O.; Clegg, A.; Turner, J.; Undersander, E.; Galuba, W.; Westbury, A.; Chang, A. X.; Savva, M.; Zhao, Y.; and Batra, D. 2021. Habitat-Matterport 3D Dataset (HM3D): 1000 Large-scale 3D Environments for Embodied AI. *ArXiv*, abs/2109.08238.

Savva, M.; Kadian, A.; Maksymets, O.; Zhao, Y.; Wijmans, E.; Jain, B.; Straub, J.; Liu, J.; Koltun, V.; Malik, J.; Parikh, D.; and Batra, D. 2019. Habitat: A Platform for Embodied AI Research. *IEEE/CVF International Conference on Computer Vision*, 9338–9346.

Song, X.; Chen, W.; Liu, Y.; Chan, V.; Li, G.; and Lin, L. 2024. Towards long-horizon vision-language navigation: Platform, benchmark and method. *arXiv preprint arXiv:2412.09082*.

Song, X.; Chen, W.; Liu, Y.; Chen, W.; Li, G.; and Lin, L. 2025. Towards long-horizon vision-language navigation: Platform, benchmark and method. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 12078–12088.

Tan, A. H.; Fung, A.; Wang, H.; and Nejat, G. 2025. Mobile Robot Navigation Using Hand-Drawn Maps: A Vision Language Model Approach. *arXiv preprint arXiv:2502.00114*.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; Rodriguez, A.; Joulin, A.; Grave, E.; and Lample, G. 2023. LLaMA: Open and Efficient Foundation Language Models. *ArXiv*, abs/2302.13971.

Wang, H.; Wang, W.; Liang, W.; Xiong, C.; and Shen, J. 2021. Structured Scene Memory for Vision-Language Navigation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8451–8460.

Wang, X.; Chen, X.; Ren, W.; Han, Z.; Fan, H.; Tang, Y.; and Liu, L. 2024a. Compensation Atmospheric Scattering Model and Two-Branch Network for Single Image Dehazing. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 8: 2880–2896.

Wang, X.; Chen, X.; Wang, F.; Xu, C.; and Tang, Y. 2023a. Image Recovery and Object Detection Integrated Algorithms for Robots in Harsh Battlefield Environments. In *Intelligent Robotics and Applications*, 575–585.

Wang, X.; Ren, W.; Chen, X.; Fan, H.; Tang, Y.; and Han, Z. 2024b. Uni-YOLO: Vision-Language Model-Guided YOLO for Robust and Fast Universal Detection in the Open World. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 1991–2000.

Wang, Z.; Li, X.; Yang, J.; Liu, Y.; Hu, J.; Jiang, M.; and Jiang, S. 2024c. Lookahead Exploration with Neural Radiance Representation for Continuous Vision-Language Navigation. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13753–13762.

Wang, Z.; Li, X.; Yang, J.; Liu, Y.; and Jiang, S. 2023b. Gridmm: Grid memory map for vision-and-language navigation. In *Proceedings of the IEEE/CVF International conference on computer vision*, 15625–15636.

Wang, Z.; Yan, H.; Wang, Z.; Xu, Z.; Wu, Z.; and Wang, Y. 2024d. Research on autonomous robots navigation based on reinforcement learning. In *2024 3rd International Conference on Robotics, Artificial Intelligence and Intelligent Control (RAIIC)*, 78–81. IEEE.

Wei, M.; Wan, C.; Yu, X.; Wang, T.; Yang, Y.; Mao, X.; Zhu, C.; Cai, W.; Wang, H.; Chen, Y.; Liu, X.; and Pang, J. 2025. StreamVLN: Streaming Vision-and-Language Navigation via SlowFast Context Modeling. arXiv:2507.05240.

Yadav, K.; Ali, Y.; Gupta, G.; Gal, Y.; and Kira, Z. 2025. FindingDory: A Benchmark to Evaluate Memory in Embodied Agents. *arXiv preprint arXiv:2506.15635*.

Yu, T.; Wu, Y.; Cui, Q.; Huang, Q.; and Yu, J. 2025. MossVLN: Memory-Observation Synergistic System for Continuous Vision-Language Navigation. *IEEE Transactions on Multimedia*.

Zhang, H.; Zantout, N.; Kachana, P.; Wu, Z.; Zhang, J.; and Wang, W. 2024a. VLA-3D: A dataset for 3D semantic scene understanding and navigation. *arXiv preprint arXiv:2411.03540*.

Zhang, J.; Cai, K.; Fan, Y.; Wang, J.; and Wang, K. 2025a. CF-VLM: CounterFactual Vision-Language Finetuning. *arXiv preprint arXiv:2506.17267*.

Zhang, J.; Cai, K.; Yang, J.; and Wang, K. 2025b. Learning Dynamics of VLM Finetuning. *arXiv preprint arXiv:2510.11978*.

Zhang, J.; Fan, Y.; Lin, W.; Chen, R.; Jiang, H.; Chai, W.; Wang, J.; and Wang, K. 2025c. GAM-Agent: Game-Theoretic and Uncertainty-Aware Collaboration for Complex Visual Reasoning. *arXiv preprint arXiv:2505.23399*.

Zhang, J.; Wang, K.; Wang, S.; Li, M.; Liu, H.; Wei, S.; Wang, Z.; Zhang, Z.; and Wang, H. 2024b. Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. *arXiv preprint arXiv:2412.06224*.

Zhang, J.; Wang, K.; Xu, R.; Zhou, G.; Hong, Y.; Fang, X.; Wu, Q.; Zhang, Z.; and He, W. 2024c. NaVid: Video-based VLM Plans the Next Step for Vision-and-Language Navigation. *ArXiv*, abs/2402.15852.

Zhang, Y.; Guo, Q.; and Kordjamshidi, P. 2024. Navhint: Vision and language navigation agent with a hint generator. Association for Computational Linguistics.

Zhao, G.; Li, G.; Chen, W.; and Yu, Y. 2024. OVER-NAV: Elevating Iterative Vision-and-Language Navigation with Open-Vocabulary Detection and StructurEd Representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16296–16306.

Zheng, D.; Huang, S.; Zhao, L.; Zhong, Y.; and Wang, L. 2024. Towards learning a generalist model for embodied navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13624–13634.

Zheng, Q.; Liu, D.; Wang, C.; Zhang, J.; Wang, D.; and Tao, D. 2025. Esceme: Vision-and-language navigation with episodic scene memory. *International Journal of Computer Vision*, 133: 254–274.