

Following Route Instructions using Large Vision-Language Models: A Comparison between Low-level and Panoramic Action Spaces

Vebjørn Haug Kåsene

University of Oslo
vebjorhk@gmail.com

Pierre Lison

Norwegian Computing Center
plison@nr.no

Abstract

Vision-and-Language Navigation (VLN) refers to the task of enabling autonomous robots to navigate unfamiliar environments by following natural language instructions. While recent Large Vision-Language Models (LVLMs) have shown promise in this task, most current VLN systems rely on models specifically designed and optimized for navigation, leaving the potential of off-the-shelf LVLMs underexplored. Furthermore, while older VLN approaches used low-level action spaces with egocentric views and atomic actions (such as "turn left" or "move forward"), newer models tend to favor panoramic action spaces with discrete navigable viewpoints. This paper investigates (1) whether off-the-shelf LVLMs (fine-tuned without architectural modifications or simulator-based training) can effectively support VLN tasks and (2) whether such models can support both low-level and panoramic action paradigms. To this end, we fine-tune the open-source model Qwen2.5-VL-3B-Instruct on the *Room-to-Room* (R2R) dataset and evaluate its empirical performance across both low-level and panoramic action spaces. The best resulting model achieves a 41% success rate on the R2R test set, demonstrating that while off-the-shelf LVLMs can learn to perform Vision-and-Language Navigation, they still lag behind models specifically designed for this task.

1 Introduction

Mobile robots deployed in real-world environments are often tasked with reaching specific locations described in natural language. For example, a robot might be instructed to "deliver a package to the office at the end of the hallway," without prior knowledge of the environment. In such cases, a human can provide guidance through route instructions such as "Walk down the hallway and take the last door to your left." To perform its task, the robot must first interpret the linguistic input provided by the human user, ground this input in its visual

perception of the environment, and execute the corresponding sequence of physical actions to reach the target location.

This problem is addressed in the field of Vision-and-Language Navigation (VLN) (Anderson et al., 2018b), which focuses on developing autonomous robotic agents that can navigate unseen environments based on natural language instructions. A common VLN benchmark and dataset is Room-to-Room (R2R) (Anderson et al., 2018b), which contains thousands of trajectory–instruction pairs, where the task is to follow natural language instructions to reach a target location. R2R is typically used in combination with the Matterport3D simulator (Anderson et al., 2018b), which simulates indoor environments reconstructed from real-world 3D scans from the Matterport3D dataset (Chang et al., 2018). The simulator represents these environments as navigation graphs, where nodes correspond to navigable locations and edges define transitions between them.

Early approaches to VLN primarily relied on RNN-based sequence-to-sequence models to encode route instructions and predict actions (Anderson et al., 2018b; Fried et al., 2018). Later work shifted toward using pre-trained transformer-based models (Vaswani et al., 2017), which offered improved language understanding and generalization (Li et al., 2019; Chen et al., 2021, 2022).

More recently, researchers have begun exploring the use of Large Language Models (LLMs) and Large Vision-Language Models (LVLMs) for VLN, using both zero-shot prompting (Zhou et al., 2024; Chen et al., 2024) and trained approaches (Zheng et al., 2024; Zhou et al., 2025). While zero-shot methods have shown promise in navigation tasks, their performance still falls short of VLN-specialized transformer-based models (Zhou et al., 2025). Most existing VLN approaches thus seek to train LLMs and LVLMs directly on VLN datasets. Although these trained approaches have achieved

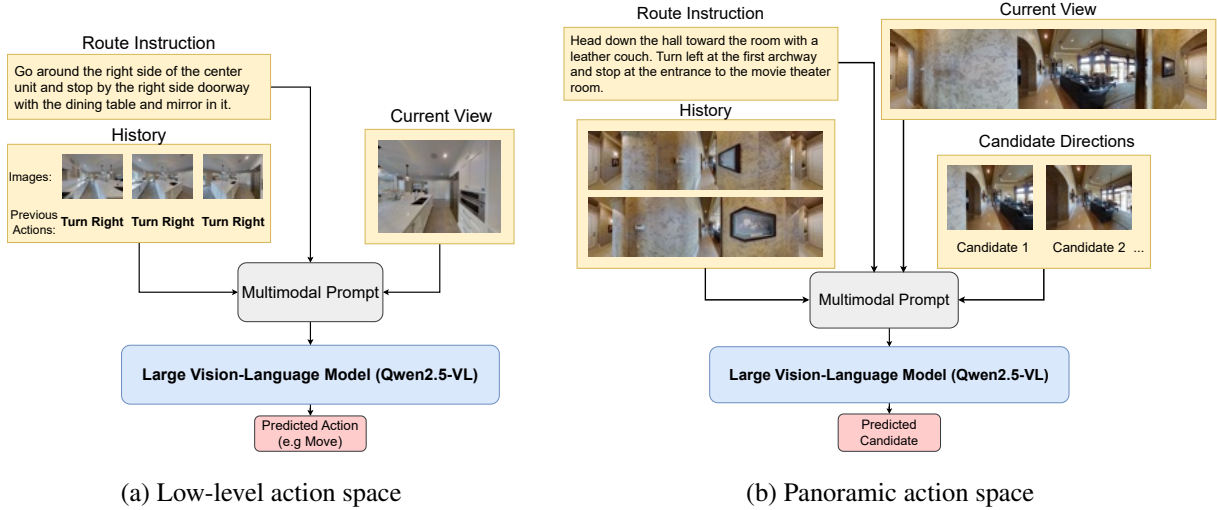


Figure 1: Overview of the approach, which is based on fine-tuning a pre-trained LVLM (Qwen2.5-VL) on the R2R dataset. The model receives as input a multimodal prompt consisting of the route instruction, navigation history, and current view, and outputs the next navigation action.

strong results, they typically rely on custom models that require either changes to the underlying neural architecture or the addition of task-specific components, such as simulators employed at training time (Zheng et al., 2024; Zhou et al., 2025). As a result, the potential of off-the-shelf LVLMs, fine-tuned for VLN without architectural changes, remains largely underexplored.

In addition, the choice of action space – i.e. the possible outputs that the model is designed to generate – has been shown to significantly affect performance (Fried et al., 2018). Early RNN-based approaches typically employed a low-level action space, where the agent observes the environment through an egocentric image and selects from a discrete set of atomic actions such as Move Forward, Turn Left, or Turn Right (Anderson et al., 2018b; Landi et al., 2019). However, low-level action spaces have largely been abandoned in recent work in favor of panoramic action spaces (Li et al., 2019; Chen et al., 2022; Zhou et al., 2024), where the agent perceives its surroundings through a 360° panoramic image and chooses among a set of navigable candidate directions, each typically corresponding to an adjacent node in the navigation graph. This shift has been shown to substantially improve performance over low-level alternatives (Fried et al., 2018). While this difference in performance has been explored in the context of RNN-based models (Fried et al., 2018; Landi et al., 2019), it has to our knowledge never been investigated for LVLM-based approaches. While

panoramic action spaces do seem to improve the navigation performance, they also assume a greater prior knowledge about the environment – such as which directions are navigable – and effectively reduce the task to a visually guided graph search (Landi et al., 2019; Krantz et al., 2020). Panoramic action spaces also depend on the availability of panoramic visual input, which in practice requires specialized robot-mounted hardware, such as panoramic or multi-camera rigs.

This paper seeks to address these knowledge gaps through experiments with a state-of-the-art LVLM, Qwen2.5-VL (Bai et al., 2025). An overview of our approach is illustrated in Figure 1. The two main contributions of this paper are:

- The evaluation of off-the-shelf LVLMs (without architectural changes or simulation-based training methods) on VLN through experiments on the R2R dataset.
- An analysis of how the choice of action space (low-level versus panoramic) affects the navigation performance.

The rest of this paper is as follows. We first review related work on Vision-and-Language Navigation and LVLMs. We then present our approach in Section 3, focusing on the fine-tuning process and the definition of possible action spaces. Section 4 then describes the experimental setup and the results obtained on the R2R dataset. Finally, Section 5 discusses those results and Section 6 concludes this paper.

2 Related Work

Large Vision-Language Models in VLN

Motivated by recent progress with LLMs and LVLMs, several studies have investigated how those models can be applied for VLN. NavGPT (Zhou et al., 2024) employs GPT-4 (OpenAI et al., 2024) in a zero-shot setting, relying on a separate model to convert visual inputs into textual descriptions. In contrast, MapGPT (Chen et al., 2024) prompts GPT-4V to perform joint reasoning over visual inputs and navigation instructions.

NaviLLM (Zheng et al., 2024) uses a frozen Vision Transformer (ViT) (Dosovitskiy et al., 2021) and models spatial relationships between different viewpoints through a trained transformer-based multi-view fusion component which produces a single visual feature for each image. NavGPT-2 (Zhou et al., 2025) uses a frozen LVLM to produce reasoning text from image-instruction pairs and fine-tunes a separate graph-based policy to predict actions and model the topological graph on the fly. Both approaches achieve state-of-the-art performance on R2R, demonstrating the potential of LLMs and LVLMs for navigation.

Action Spaces in VLN

Early approaches to VLN employ a low-level action space where the agent perceives the world through an egocentric image at each step and predicts actions such as Move Forward or Turn Right (Anderson et al., 2018b; Wang et al., 2018; Fried et al., 2018). Fried et al., 2018 introduce panoramic action space for VLN. Instead of receiving an egocentric image as input, the model is provided with a panorama comprised of 36 images at different angles. The images closest to the center of an adjacent node are considered as candidate views. Instead of predicting low-level actions, the agent selects between which of these views to navigate to. Using an LSTM (Hochreiter and Schmidhuber, 1997) seq-2-seq model, they observe a 12% performance increase on R2R when going from low-level to panoramic action space.

Although there is little recent work on low-level action spaces in discrete environments (VLN-DE), it remains the most common approach for VLN in continuous environments (VLN-CE) (Krantz et al., 2020; Zhang et al., 2024) where agents are tasked with navigating environments not constrained by a predefined navigation graph. In this work, we focus on VLN in discrete environments.

Modality alignment in LVLMs

Modern Large Vision-Language Models (LVLMs) typically comprise three core components: a vision encoder (e.g., a Vision Transformer (Dosovitskiy et al., 2021)), a cross-modal projector, and a text encoder (e.g., an LLM) (Bai et al., 2025). The role of the cross-modal projector is to align the visual features produced by the vision encoder with the latent space of the LLM.

Laurençon et al. (2024) investigate key design choices in building LVLMs and identify two prevalent architectural paradigms for vision-language alignment. The first is the *cross-attention architectures*, in which visual features are injected at different layers within the LLM, one example of such a model is Flamingo (Alayrac et al., 2022). The second is the *fully autoregressive architectures* where the output of the vision encoder is projected into the input space of the LLM and concatenated with the sequence of text embeddings as a multi-modal prompt (Zhu et al., 2023; Li et al., 2023). The model used in this study, Qwen2.5-VL, follows this fully autoregressive design.

3 Method

3.1 Problem Formulation

We adopt the standard VLN in discrete environments (VLN-DE) setup (Anderson et al., 2018b; Fried et al., 2018; Chen et al., 2022), where the environment is modeled as an undirected graph $G = \{V, E\}$. The nodes $V = \{v_i\}_{i=1}^K$ represent K navigable locations while the edges E constitute navigation paths between them. We then formulate the problem of following route instructions in a graph-based environment as follows: given a natural language route instruction $W = \{w_1, w_2, \dots, w_L\}$, the agent is tasked with following the instruction to reach the goal location. At each time step t , the agent receives a visual observation O_t , maintains a history context H_t , and is provided with auxiliary signals such as the cumulative distance traveled $d_t \in \mathbb{R}$ and the current step number t . The specific formulation of the agent’s input and output depends on the underlying action space, as described below.

3.2 Low-level Action Space

In the low-level action space, the agent perceives its environment through an egocentric image O_t at each step. It maintains a historical context $H_t = \{(O_1, a_1), (O_2, a_2), \dots, (O_{t-1}, a_{t-1})\}$

where O_{t-1} and a_{t-1} are the image and action from the previous step, respectively. Additionally, the agent is provided with a set of low-level actions $U_t = \{u_1, u_2, \dots, u_k\}$ that represent the actions allowed at step t , given the physical constraints of the environment (e.g., the agent cannot move forward if directly facing a wall). The agent predicts the next action a_t by estimating the probability:

$$P(a_t \mid W, O_t, H_t, d_t, t, U_t) \quad (1)$$

The low-level action space used in this work consists of four discrete actions:

- **Move:** moves forward to the node closest to the center of the current field of view.
- **Left, Right:** rotate the agent by 30° in the respective direction.
- **Stop:** signals that the agent believes it has reached the goal.

A limitation of this setup is that navigation is constrained to a discrete graph of nodes. The Move action advances the agent to the node most centered in its current field of view, but this target is not necessarily aligned with the agent’s heading. As a result, the agent may appear to move sideways, which can lead to non-intuitive trajectories. To mitigate this, an automatic reorientation step, referred to as Automatically Turn Towards Node, is applied before each Move action. Although this reorientation is not part of the learnable action space, both the resulting observation and action are included in the agent’s history. This adjustment allows us to evaluate whether explicitly aligning the agent’s heading with its movement direction improves navigation performance.

3.3 Panoramic Action Space

With the panoramic action space, the agent perceives the environment through a 360° panoramic image O_t at each step, aligned with its current heading. The agent maintains a history of panoramic views $H_t = \{O_1, \dots, O_{t-1}\}$ and selects from a set of navigable candidate views $C_t = \{c_1, \dots, c_k\}$. Each candidate c_i includes an image, a relative heading $\theta_i \in [-180^\circ, 180^\circ]$, and an associated travel distance $\delta_i \in \mathbb{R}_{\geq 0}$. The task for each step is to predict the correct candidate direction c_t :

$$P(c_t \mid W, O_t, H_t, d_t, t, C_t) \quad (2)$$

Similarly to low-level actions, the episode concludes when the agent predicts the Stop action.

The panoramic image is centered on the agent’s current heading, while each candidate view is a standard perspective image oriented directly toward a navigable direction. Candidate views are sorted from left to right based on their relative angle to the panoramic center, with the leftmost candidate assigned index 0 and the rightmost index $K - 1$.

At each step, the model predicts a token corresponding to one of the candidate indices (from 0 to $K-1$) or the Stop action. Unlike traditional panoramic setups (Fried et al., 2018; Zheng et al., 2024), where candidate views are extracted from within the panorama itself, this approach treats the panorama and candidate views as separate inputs. This design, motivated by memory limitations, reduces the number of input images per step. See Appendix A for an illustrative example.

3.4 Action selection

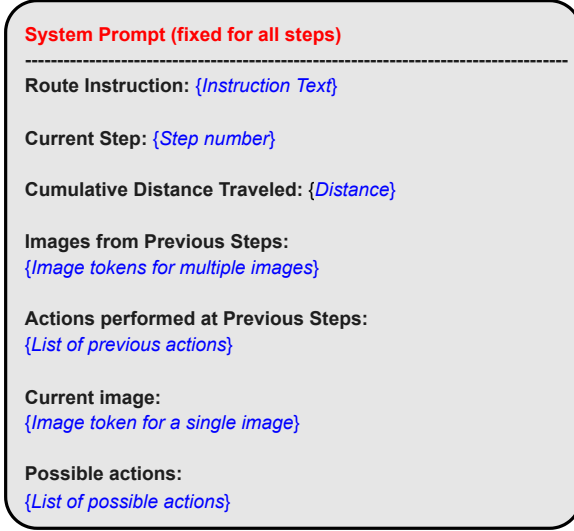
To select the next action to perform, the model receives a structured multi-modal prompt that encodes the current state, including the instruction, visual input, and auxiliary information such as step number and distance traveled. These prompts follow a fixed schema shown in Figure 2. Inference is performed greedily, selecting the most probable action at each step without backtracking.

In addition to the dynamic input state, each prompt includes a static system prompt that explains the task and describes the individual input fields. The system prompt is fixed and specific to each action space, and remains unchanged throughout training and evaluation. The full system prompts are included in Appendix A

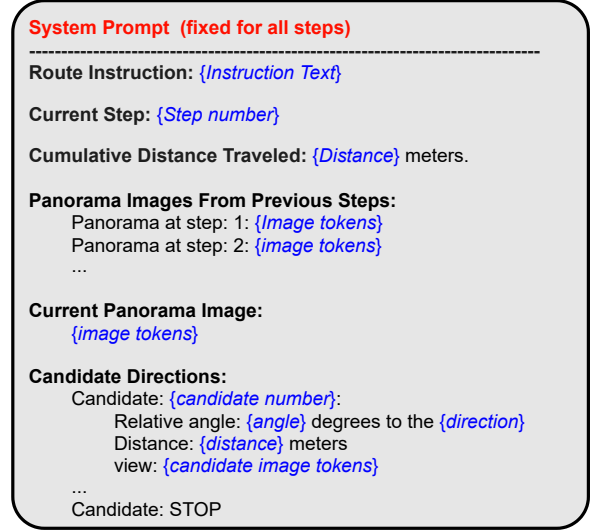
3.5 Fine-tuning

The LVLMS are fine-tuned through behavior cloning, where the model learns to imitate expert demonstrations. At each time step t , the model receives a multimodal prompt x_t represents the current state, and is trained to predict the expert action a_t as a token from its own vocabulary. The training objective minimizes the total negative log-likelihood of the expert actions over the entire episode. Gradients are accumulated across all time steps in an episode, and the weights are updated at the end of each episode.

Unlike many recent VLN approaches (Chen et al., 2021; Zhou et al., 2025; Anderson et al.,



(a) Low-level action space prompt schema



(b) Panoramic action space prompt schema

Figure 2: Prompt schemata for low-level and panoramic action spaces.

2018b), our approach does not therefore rely on reinforcement learning or student forcing, but simply fine-tunes the LVLM model on the basis of expert routes. A key advantage of this approach is the fact that it can be applied without access to a simulator at training time.

4 Experiments

The proposed approach was evaluated on the Room-to-Room (R2R) dataset using both *offline* and *on-line* evaluation modes. The offline mode assesses the model’s ability to follow expert trajectories, whereas the online mode evaluates its performance when navigating autonomously within the Matterport3D simulation environment.

4.1 Dataset

The Room-to-Room dataset (Anderson et al., 2018b) contains 21,567 English route instructions corresponding to 7,189 trajectories across 90 environments. Each ground truth trajectory is a sequence of nodes in a Matterport3D environment. Each trajectory has 3 corresponding instructions.

The dataset is split into four subsets: *train* (61 environments), *val seen* (56 environments overlapping with train), *val unseen* (11 environments), and *test* (18 environments). Performance is evaluated on the val unseen and test splits. All splits are preprocessed to convert ground truth trajectories into sequences of actions.¹

¹For the alternative low-level action spaces experiments, the models were trained on a subset consisting of the first

4.2 Evaluation Metrics

Online, the models are evaluated using standard VLN metrics (Anderson et al., 2018b). **Navigation Error (NE)** is the average walkable distance between the agent’s final location and the goal location in meters. An episode is considered successful if $NE \leq 3$ m and the last predicted action is Stop. **Path Length (PL)** is the average path length (in meters). **Oracle Success Rate (OSR)** measures the percentage of episodes in which the agent was within 3 meters of the goal at any point during the navigation episode. **Success Rate (SR)** is the percentage of episodes that are successful. **Success Weighted by Path Length (SPL)** (Anderson et al., 2018a) combines SR with path length, penalizing unnecessarily long paths. **Coverage Weighted by Length Score (CLS)** (Jain et al., 2019) measures how well the agent’s predicted path follows the route instruction, penalizing paths that deviate from the ground truth path.

For the offline evaluation, the reported metrics are **Accuracy** the proportion of actions correctly predicted by the model; **Macro F1**, the unweighted mean F1 score computed across all action classes; and **Conservative Success Rate (CSR)**, the percentage of episodes in which all actions are identical (from start to finish) to the actions selected by the expert. For offline evaluation, we use the third instruction from each trajectory.

1,955 trajectories in R2R’s train split.

4.3 Implementation

Model

We experimented with two distinct Large Vision-Language Models (LVLMs): Qwen2-VL-2B-Instruct (Wang et al., 2024) and Qwen2.5-VL-3B-Instruct (Bai et al., 2025). Qwen2.5-VL-3B is the larger of the two and is pre-trained on 4 trillion tokens, compared to 1.2 trillion tokens for Qwen2-VL-2B. As our experiments showed that Qwen2.5 consistently provided superior performance compared to Qwen2 in both offline and online metrics on the validation dataset, we only provide evaluation results obtained for Qwen2.5.

During fine-tuning, the vision encoder and the cross-modal projection layer are kept frozen, as preliminary experiments indicated that tuning only the LLM led to improved performance.²

Simulator

The Matterport3D simulator (MP3D) is used for evaluation and for generating the preprocessed training data. In MP3D, the agent’s field of view is determined by the image resolution and the vertical field of view (VFOV). This work uses an image resolution of 640×480 for egocentric and candidate images. The VFOV is set to 105° to allow the agent to perceive a broader visual context. This is substantially larger than the VFOV used in prior work, which typically ranges from 60° (Anderson et al., 2018b; Fried et al., 2018) to 75° (Krantz et al., 2020). Panoramic images are constructed by stitching together three egocentric views captured while rotating the agent in place.

Training

All models are fine-tuned with a batch size of 1, a learning rate of $1e-5$, and a weight decay of 0.1. A linear learning rate scheduler is used with warmup over the first 10% of training steps. FlashAttention (Dao et al., 2022) is enabled, and training is performed in bfloat16 precision. Input images are resized to half their original size to accommodate GPU memory constraints.³ Experiments were conducted on a single NVIDIA A100 80GB GPU. Models are fine-tuned for 1 epoch across all instructions, corresponding to 3 total passes over the unique paths (as each path in R2R is associated

²The trained models are publicly available at <https://huggingface.co/Vebbern> for reproducibility.

³meaning 320×240 for candidates and egocentric views, and 960×240 for panoramic views

Model	Accuracy $\uparrow$	Macro F1 $\uparrow$	CSR $\uparrow$
Val seen:			
Qwen2.5-VL-low	0.73	0.74	0.04
Qwen2.5-VL-pano	0.73	0.61	0.16
Val unseen:			
Qwen2.5-VL-low	0.73	0.73	0.03
Qwen2.5-VL-pano	0.73	0.62	0.15

Table 1: Offline evaluation results on the seen and unseen R2R validation sets.

with three distinct route instructions).⁴

4.4 Results

Offline evaluation Table 1 presents the offline evaluation results after fine-tuning the Qwen2.5 model on the full training set of R2R. In terms of accuracy, the panoramic and low-level models score similarly. The low-level model has a higher macro F1 score, which could be explained by the larger number of actions of panoramic models (up to 12 candidate views). However, the panoramic model has a significantly higher conservative success rate (CSR) than the low-level one. Qwen2.5-VL-pano achieves a CSR of 15% on val unseen, compared to a mere 3% CSR for Qwen2.5-VL-low.

Online evaluation Table 2 compares our results with state-of-the-art (SOTA) approaches on R2R using single-run greedy search (i.e., no pre-exploration). Results are shown for both panoramic and low-level action space.

The model fine-tuned for low-level action spaces, Qwen2.5-VL-low, achieves a success rate (SR) of 26% on the test set, outperforming the original R2R baseline (Seq2Seq, 21% SR) and Speaker-Follower (SF) without panoramic action (25% SR on val unseen). However, it still lags behind the LSTM-based DCF model of (Landi et al., 2019), which reached 35% SR, despite being substantially smaller in size. However, Qwen2.5-VL-low is less prone to overfitting to training environments, as reflected in the smaller SR gap between val seen and unseen (35% vs. 27%) compared to DCF, which drops from 58% to 34% on val unseen.

The panoramic model, Qwen2.5-VL-pano, achieves a 41% SR on the R2R test set. This outperforms all low-level models as well as panoramic approaches such as Speaker-Follower (36% on val unseen) and NavGPT (Zhou et al., 2024) (34% on

⁴The source code is available at <https://github.com/Vebjorhk/masters-thesis-VLN>.

	Val Seen					Val Unseen					Test (Unseen)				
	PL	NE↓	OSR↑	SR↑	SPL↑	PL	NE↓	OSR↑	SR↑	SPL↑	PL	NE↓	OSR↑	SR↑	SPL↑
Human	-	-	-	-	-	-	-	-	-	-	11.85	1.61	90	86	76
Low-Level															
Seq2Seq (2018b)	11.33	6.01	53	39	-	8.39	7.81	28	22	-	8.13	7.85	27	21	-
SF (2018)	-	4.28	60	47	-	-	5.75	33	25	-	-	-	-	-	-
RPA(2018)	8.46	5.56	53	43	-	7.22	7.65	32	25	-	9.15	7.53	33	25	-
DCF (2019)	-	3.96	73	58	51	-	6.52	43	34	29	9.81	6.55	45	35	31
Panoramic															
SF (2018)	-	3.36	73	66	-	-	6.62	45	36	-	-	-	-	-	-
PRESS (2019)	10.35	3.09	-	71	67	10.06	4.31	-	59	55	10.52	4.53	63	57	53
VLN $\odot$ BERT (2021)	11.13	2.90	-	72	68	12.01	3.93	-	63	57	12.35	4.09	-	63	57
HAMT (2021)	11.15	2.51	-	76	72	11.46	2.29	-	66	61	12.27	3.93	-	65	60
DUET (2022)	-	-	-	-	-	13.94	3.31	-	72	60	14.73	3.65	-	69	59
NavGPT (2024)	-	-	-	-	-	-	-	-	-	-	11.45	6.46	42	34	29
NaviLLM (2024)	-	-	-	-	-	-	-	-	-	59	-	-	-	-	60
NavGPT-2 (2025)	14.13	2.84	83	74	63	14.01	2.98	84	74	61	14.74	3.33	80	72	60
Qwen2.5-VL-low	10.27	7.14	41	35	32	10.50	7.84	34	27	24	10.59	7.99	34	26	24
Qwen2.5-VL-pano	9.98	5.69	56	50	47	9.83	6.65	46	38	35	9.96	6.53	50	41	38

Table 2: Comparison of panoramic and low-level models with state-of-the-art performance using single-run greedy search. CLS is not reported for R2R test set. Models introduced in this work are shown in bold.

Models	PL	NE↓	OSR↑	SR↑	SPL↑	CLS↑
Val Unseen:						
105-VFOV	10.17	7.87	0.34	0.25	0.23	0.45
82-VFOV	9.9	7.87	0.32	0.25	0.23	0.44
No-Adjust	10.72	7.7	0.38	0.29	0.26	0.44

Table 3: Online results on R2R val unseen for alternative definitions of the low-level action space.

test). However, this model falls short of more recent task-specific panoramic approaches such as NaviLLM (Zheng et al., 2024) (60% SPL) and NavGPT-2 (Zhou et al., 2025) (72% SR).

Alternative low-level action spaces We also explored alternative configurations for the low-level action space. Specifically, we assessed the impact of (1) disabling the Automatically Turn Towards Node action, and (2) reducing the vertical field of view (VFOV) from 105° to a narrower 82°.

Table 3 presents the performance on the val unseen split for those two alternatives. The difference between 82° and 105° VFOV is minimal, with only slight improvements in CLS and OSR scores for the 105° configuration. However, removing the adjustment action leads to a noticeable performance gain: the No-Adjust model achieves a SR of 29%, compared to 25% for the default. This suggests that explicitly facing the next node before movement is often unnecessary for effective navigation.

5 Discussion

Fine-tuning off-the-shelf LVLMS for R2R

The results indicate that fine-tuning off-the-shelf LVLMS such as Qwen2.5-VL on the R2R task fails to yield strong performance, despite the fact that such models are significantly larger than older, VLN-specific architectures such as PRESS (Li et al., 2019), DUET (Chen et al., 2022), and HAMT (Chen et al., 2021). It is difficult to pinpoint the exact source of this performance gap, though our use of behavior cloning – rather than optimization through student forcing and/or reinforcement learning, as done by e.g. (Chen et al., 2021; Zhou et al., 2025) – may be a contributing factor.

Compared to NaviLLM (Zheng et al., 2024) and NavGPT-2 (Zhou et al., 2025), which are the two approaches most similar to this work, a key difference becomes apparent. In the Qwen2.5-VL-low model, each input image is encoded and then fed directly into the LLM, which is solely responsible for interpreting the route instruction, modeling spatial relationships between images, and predicting actions. While Qwen2.5-VL reduces visual token count through patch merging, it does not incorporate any explicit mechanisms for modeling spatial structure between images before they are fed to the LLM. In contrast, NaviLLM (Zheng et al., 2024) includes a transformer-based module that explicitly captures the spatial relationships be-

Split	Avg. steps per path (low-level)	Avg. steps per path (panoramic)
train	12.88	6.00
val seen	12.85	6.07
val unseen	13.40	5.97

Table 4: Average number of steps (actions) for panoramic and low-level variants of R2R.

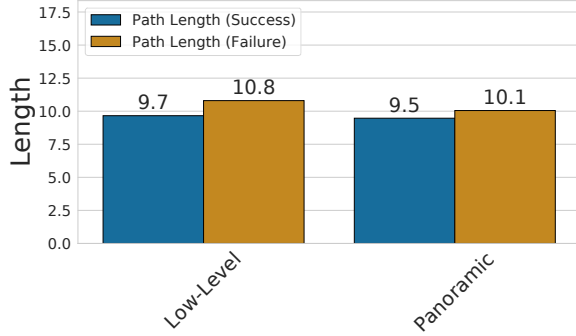


Figure 3: Avg. path length (meters) on R2R val unseen.

tween panoramic images before it is fed as input to the LLM. NavGPT-2 (Zhou et al., 2025) takes this further by using a separate graph-based policy network to model viewpoint connectivity and predict actions, while delegating route-level reasoning to the LLM. These design differences may help explain at least part of the observed performance gap: relying solely on the LLM for spatial reasoning and control can be challenging – especially for longer paths – compared to models that explicitly encode spatial structure. Prior work also suggests that reducing visual tokens benefits non-OCR tasks (Laurençon et al., 2024). Both Nav-iLLM and NavGPT-2 use significantly fewer visual tokens than Qwen2.5-VL (Bai et al., 2025).

Panoramic vs. low-level action space The panoramic models consistently outperform low-level ones, which aligns with previous findings by Fried et al. (2018), although the performance gap in our setup is slightly larger (16% vs. 12% SR) showing that the panoramic approach leads to better results for LVLMs as well. One plausible explanation for the performance gap is that low-level action sequences are, on average, twice as long as those in the panoramic setting (Table 4). As shown in Figure 3, both model types tend to perform better on shorter trajectories. This suggests that longer sequences in the low-level setting increase the diffi-

culty of the task, as they provide more opportunities for errors to accumulate and make recovery more challenging. This is further supported by the noticeably higher Conservative Success Rate (CSR) for panoramic models (Table 1), indicating they are more likely to keep the agent on the correct path. In contrast, low-level models are more prone to errors due to the increased number of decision points, making it harder to recover once the agent deviates from the intended path.

The extent to which the additional visual information provided by panoramic images contributes to improved performance remains somewhat unclear. Panoramic observations may benefit the agent by reducing the need for physical reorientation to perceive important landmarks. Low-level action spaces may also place greater demands on spatial and temporal reasoning abilities: the agent must not only ground instructions in the visual context but also anticipate when certain actions should be executed – such as recognizing that a given action may only occur after completing several turns.

6 Conclusion

This work focused on the use of off-the-shelf Large Vision-Language Models (LVLMs) for Vision-and-Language Navigation (VLN) tasks. More precisely, we investigated how such models could be fine-tuned directly from expert routes, without modifying the model’s underlying architecture or relying on online approaches that necessitate the use of a simulator at training time. The performance of this approach was assessed through experiments on the R2R dataset and explored using both panoramic and low-level action spaces.

The best performing model, fine-tuned from Qwen2.5-VL, achieved a success rate (SR) of 41% on the R2R dataset. Our results suggest that simply fine-tuning LVLMs remains insufficient to reach state-of-the-art performance on navigation tasks. Furthermore, we find that the performance gap between low-level and panoramic action spaces persists even with larger and more powerful models, with a 16% difference in SR on the R2R test set in favor of the panoramic setup.

A promising topic for future work is the systematic study of off-the-shelf LVLMs on the R2R dataset. Evaluating a broader range of models beyond Qwen2 and Qwen2.5 could help identify which architectural choices lead to better navigation performance. Additionally, a more focused

investigation of the panoramic action space is warranted – particularly through ablation studies that isolate the effect of including a panoramic view, and systematically vary the field of view to understand its impact on performance. We also encourage future work to further explore the low-level action space for more recent approaches, including adapting it to existing state-of-the-art methods such as NaviLLM (Zheng et al., 2024) and NavGPT-2 (Zhou et al., 2025) and comparing the performance to panoramic action space.

Limitations

We acknowledge several limitations in this work. Most importantly, the fine-tuning approach is limited to behavior cloning, and did not include the use of VLN training techniques such as student forcing or reinforcement learning, potentially limiting direct comparability with prior work that leverages these strategies. For evaluation, we set up a web API to communicate with the machine running the simulator remotely. However, this introduced an additional limitation: the need for network calls made simulator evaluation significantly more time-consuming. As a result, we restricted evaluation of alternative setups to only the first third of the route instructions.

GPU memory limitations restricted training to a batch size of 1. To further reduce memory usage, we deviated from the standard panoramic action format used in many VLN approaches (Fried et al., 2018; Li et al., 2019; Hong et al., 2021), where the model receives a set of discrete view images (typically 36) and selects candidate views from among them. Instead, we preprocessed the full panorama as a single image and treated candidate views as separate, independent inputs. This setup reduces granularity, introduces visual artifacts, and limits comparability to prior benchmarks.

Finally, we note that Room-to-Room (R2R) contains only English-language route instructions, which limits the applicability of our approach to English-only scenarios. While multilingual VLN datasets have been proposed – such as Room-across-Room (RxR) (Ku et al., 2020) – our current experiments do not address multilingual aspects.

Ethics Statement

This work investigated the use of off-the-shelf Large Vision-Language Models (LVLMs) for Vision-and-Language Navigation (VLN) tasks. All

models used in this study are open-source and publicly available. The dataset employed, Room-to-Room (R2R), is a widely used benchmark in the VLN community and does not contain personally identifiable information. We do not foresee any direct ethical concerns related to the methods, data, or potential applications of this research. Our study adheres to the ACL Code of Ethics.

Acknowledgments

This work is based on research conducted as part of the first author’s Master’s thesis at the University of Oslo in 2025. Pierre Lison’s work was funded by the SFI NorwAI, (Center for Research-based Innovation, 309834). All figures and visualizations derived from the Matterport3D dataset are used in accordance with the Matterport Terms of Service: https://kaldir.vc.in.tum.de/matterport/MP_TOS.pdf.

References

- Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikołaj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. 2022. *Flamingo: a Visual Language Model for Few-Shot Learning*. In *Advances in Neural Information Processing Systems*, volume 35, pages 23716–23736.
- Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Motlaghi, Manolis Savva, and Amir R. Zamir. 2018a. *On Evaluation of Embodied Navigation Agents*. ArXiv:1807.06757.
- Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sunderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. 2018b. *Vision-and-Language Navigation: Interpreting Visually-Grounded Navigation Instructions in Real Environments*. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3674–3683.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. 2025. *Qwen2.5-VL Technical Report*. ArXiv:2502.13923.

- Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2018. [Matterport3D: Learning from RGB-D data in indoor environments](#). In *Proceedings - 2017 International Conference on 3D Vision, 3DV 2017*, pages 667–676.
- Jiaqi Chen, Bingqian Lin, Ran Xu, Zhenhua Chai, Xiaodan Liang, and Kwan-Yee Wong. 2024. [MapGPT: Map-Guided Prompting with Adaptive Path Planning for Vision-and-Language Navigation](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9796–9810, Bangkok, Thailand. Association for Computational Linguistics.
- Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. 2021. [History Aware Multimodal Transformer for Vision-and-Language Navigation](#). In *Advances in Neural Information Processing Systems*, volume 34, pages 5834–5847. Curran Associates, Inc.
- Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. 2022. [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*, pages 16537–16547.
- Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. [FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 16344–16359.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. [An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale](#). In *International Conference on Learning Representations*.
- Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. 2018. [Speaker-Follower Models for Vision-and-Language Navigation](#). In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. [Long Short-term Memory](#). *Neural computation*, 9:1735–80.
- Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. 2021. [VLNBERT: A Recurrent Vision-and-Language BERT for Navigation](#). In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1643–1653.
- Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. 2019. [Stay on the Path: Instruction Fidelity in Vision-and-Language Navigation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1862–1872, Florence, Italy. Association for Computational Linguistics.
- Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. 2020. [Beyond the Nav-Graph: Vision-and-Language Navigation in Continuous Environments](#). In *Computer Vision – ECCV 2020*, pages 104–120, Cham. Springer International Publishing.
- Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. 2020. [Room-Across-Room: Multilingual Vision-and-Language Navigation with Dense Spatiotemporal Grounding](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4392–4412, Online. Association for Computational Linguistics.
- Federico Landi, Lorenzo Baraldi, Massimiliano Corsini, and Rita Cucchiara. 2019. [Embodied Vision-and-Language Navigation with Dynamic Convolutional Filters](#). In *Proceedings of the British Machine Vision Conference*.
- Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. 2024. [What matters when building vision-language models?](#) In *Advances in Neural Information Processing Systems*, volume 37, pages 87874–87907.
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. [BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models](#). In *Proceedings of the 40th International Conference on Machine Learning*, pages 19730–19742.
- Xiujun Li, Chunyuan Li, Qiaolin Xia, Yonatan Bisk, Asli Celikyilmaz, Jianfeng Gao, Noah A. Smith, and Yejin Choi. 2019. [Robust Navigation with Language Pretraining and Stochastic Sampling](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1494–1499, Hong Kong, China. Association for Computational Linguistics.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully

- Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Felipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, C. J. Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lillian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. [GPT-4 Technical Report](#). ArXiv:2303.08774.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is All you Need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024. [Qwen2-VL: Enhancing Vision-Language Model’s Perception of the World at Any Resolution](#). ArXiv:2409.12191.
- Xin Wang, Wenhan Xiong, Hongmin Wang, and William Yang Wang. 2018. [Look Before You Leap: Bridging Model-Free and Model-Based Reinforcement Learning for Planned-Ahead Vision-and-Language Navigation](#). In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 37–53.
- Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. 2024. [NaVid: Video-based VLM Plans the Next Step for Vision-and-Language Navigation](#). In *Robotics: Science and Systems*.
- Duo Zheng, Shijia Huang, Lin Zhao, Yiwu Zhong, and Liwei Wang. 2024. [Towards Learning a Generalist Model for Embodied Navigation](#). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13624–13634.
- Gengze Zhou, Yicong Hong, Zun Wang, Xin Eric Wang, and Qi Wu. 2025. [NavGPT-2: Unleashing Navigational Reasoning Capability for Large Vision-Language Models](#). In *Computer Vision – ECCV 2024*, pages 260–278, Cham. Springer Nature Switzerland.
- Gengze Zhou, Yicong Hong, and Qi Wu. 2024. [NavGPT: Explicit Reasoning in Vision-and-Language Navigation with Large Language Models](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 7641–7649.
- Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. [MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models](#). In *The Twelfth International Conference on Learning Representations*.

A Appendix

You are a robot which follows route instructions step-by-step to reach a destination.

At every step, you will receive:

1. Route Instruction: the instruction to follow.
2. Current Step: The step number you are currently on in the overall route.
3. Cumulative Distance Traveled: The total distance (in meters) you have moved from the starting point up to your current position.
4. Observations from previous steps (if available), including:
 - Images captured at previous steps.
 - Actions performed at previous steps.
5. Current image: An image showing the robots present view.
6. Possible actions: The set of available actions for this step.

Actions and their definitions:

- Right: Rotates 30 degrees to the right.
- Left: Rotates 30 degrees to the left.
- Move: Moves you forward in your current direction of view.
- Stop: Choose this action when you think you have reached the goal or the end of the navigation path.

Important Notes

- Choose only one action at a time, using only the predefined actions listed in the 'Possible Actions' field.
- The environment is graph-based, meaning movement occurs between discrete nodes rather than continuous space.
- Automatically Turn Towards Node: When you move forward, the camera is automatically adjusted to center on the next node in the graph-based environment. This is handled separately and does not require prediction. Your responsibility is to decide movement and rotation based on the provided inputs.

Your task is to predict the most appropriate next action at each step based on the given information.

Figure 4: System prompt for low-level action space

You are a robot which follows route instructions step-by-step to reach a destination. At every step, you will receive:

1. Route Instruction: the instruction to follow.
2. Current Step: The step number you are currently on in the overall route.
3. Cumulative Distance Traveled: The total distance (in meters) you have moved from the starting point up to your current position.
4. Panorama Images from Previous Steps: If available, these images provide context about where you have been. Use them to understand your past movements and to identify which parts of the current route instruction are most relevant to your current step.
5. Current Panorama Image: A 360-degree panoramic image of your current surroundings. The center of the image represents your current forward direction.
6. Candidate Directions: A list of possible directions to move.
Each candidate includes:
 - Relative angle: The direction relative to your forward orientation (e.g., '30° left' or '45° right').
 - Distance: The distance (in meters) to the next possible location in that direction.
 - A view (image): What you would see if you move in that direction.
7. STOP Candidate: This is always available and must only be selected when you are certain you have reached the final destination as described in the route instruction.

Your task:

Using the provided inputs, analyze and select the one candidate direction that best matches the route instruction and ensures you stay aligned with the intended path.

Figure 5: System prompt for panoramic action space

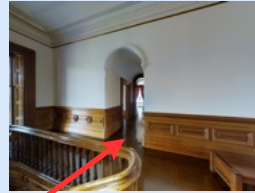
With adjustment



Turn Left



Automatically Turn
Towards Node



Move Forward



Without Adjustment



Turn Left



Move Forward

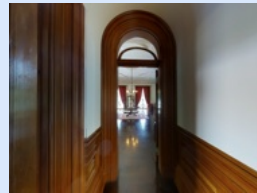


Figure 6: Figure illustrating the Automatically Turn Towards Node step.

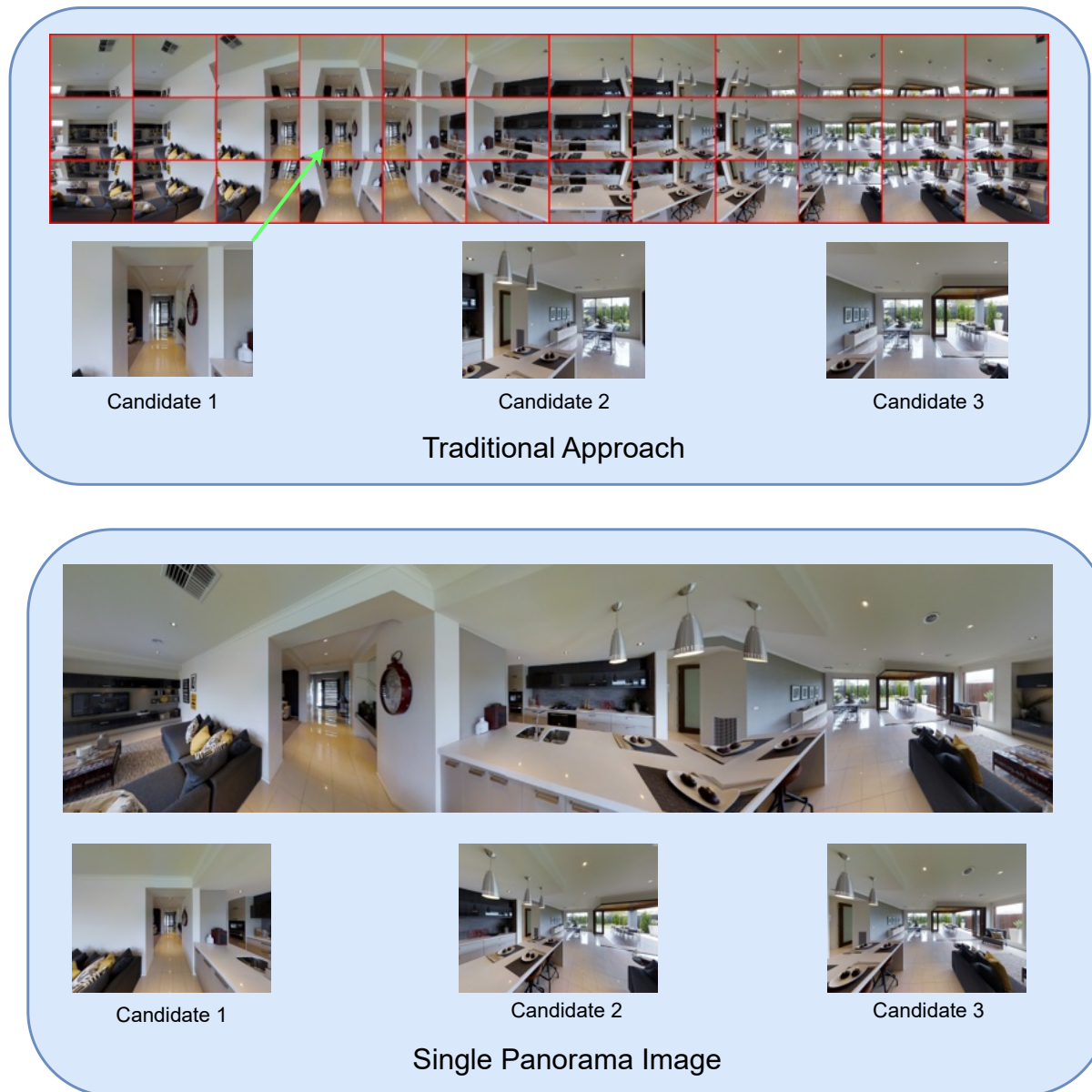


Figure 7: Figure illustrating the difference between the traditional panoramic approach and our implementation