LaF-GRPO: In-Situ Navigation Instruction Generation for the Visually Impaired via GRPO with LLM-as-Follower Reward

Anonymous ACL submission

Abstract

Navigation instruction generation for visually impaired (VI) individuals (NIG-VI) is critical yet relatively underexplored. This study, hence, focuses on producing precise, in-situ, step-by-step navigation instructions that are practically usable by VI users. Concretely, we propose LaF-GRPO (LLM-as-Follower GRPO), where an LLM simulates VI user responses to generate rewards guiding the Vision-Language Model (VLM) post-training. This enhances instruction usability while reducing costly real-world data needs. To facilitate training and testing, we introduce NIG4VI, a 27k-sample open-sourced benchmark. It provides diverse navigation scenarios with accurate spatial coordinates, supporting detailed, open-ended in-situ instruction generation. Experiments on NIG4VI show the effectiveness of LaF-GRPO by quantitative metrics (e.g., Zero-(LaF-GRPO) boosts BLEU +14%; SFT+(LaF-GRPO) METEOR 0.542 vs. GPT-4o's 0.323) and yields more intuitive, safer instructions. Code and benchmark are available at https://github.com/instruction-generation/anonymous-llm-as-follower.

1 Introduction

The Visually Impaired (VI) community, comprising approximately 2.2 billion individuals globally with partial or complete blindness, underscores the significant need for effective assistive technologies.[1] Enhancing their quality of life through Visually Impaired Assistance (VIA) has motivated extensive research, e.g., VIALM (Zhao et al., 2024) and WalkVLM (Yuan et al., 2025). This paper specifically focuses on Navigation Instruction Generation for VI users (**NIG-VI**), a VIA sub-area. Navigation Instruction Generation (NIG) was initially conceptualized for general embodied agents, producing high-level trajectory plans. Yet, NIG-VI, being *people-centered*, operates under fundamentally different constraints. As shown in Figure 1, effective

[1] https://www.who.int/publications/i/item/9789241516570

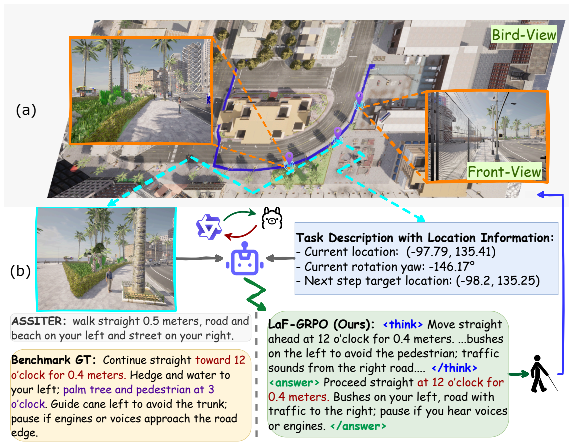


Figure 1: A NIG4VI sample. (a) The bird's-eye view map with waypoints and corresponding front-view images. (b) The system takes front-view images, task descriptions, and position data (in the blue box) as input to generate navigation instructions with the benchmark output (in the yellow box) and our model output (in the green box). The grey box indicates a non-VI navigation.

NIG-VI systems must generate *in-situ step-level instructions* that (1) integrate non-visual sensory cues (auditory/tactile landmarks, surface textures) (2) provide accurate directional and distance guidance to compensate for lack of visual referencing, and (3) adapt to urban obstacles - all within map-coordinate systems while ensuring walking safety.

Early attempts such as ASSISTER (Huang et al., 2022) laid the initial groundwork in this field, yet suffered from architectural limitations of BERT-based systems (Devlin et al., 2019). The advent of Vision-Language Models (VLMs) introduces new opportunities through their multimodal understanding and generation capabilities. RL-based post-training methods like GRPO (DeepSeek-AI, 2025) further enhance reasoning abilities, enabling VLMs to align with NIG-VI that demands people-centered guidance. However, most existing approaches rely on large-scale parallel data for fine-tuning, which can be costly and fail to incorporate interactive user feedback essential for people-centered guidance.

To bridge this gap, we propose **LLM-as-Follower GRPO (LaF-GRPO)**—the first GRPO-based framework for the NIG-VI task, featuring two novel components: (1) an LLM that simulates VI user responses to navigation instructions by interpreting their likely actions, and (2) a VLM post-training procedure for instruction generation, guided by LLM-as-Follower reward. LaF-GRPO mitigates the need for costly VI user trials while ensuring instruction usability with human-in-the-loop navigation simulation. Also, viewing the scarcity of VI navigation benchmarks, we construct **NIG4VI** - a comprehensive VI navigation instruction benchmark featuring 27k samples for simulation experiments. Fully open-sourced with granular spatial metadata, NIG4VI enables the generation of detailed, open-ended in-situ instructions.

We then experiment with LaF-GRPO on NIG4VI and find: (1) Qwen2.5-VL models trained with Zero-(LaF-GRPO) outperforms the Zero-Shot baseline, achieving superior scores across diverse metrics. (2) Qwen2.5-VL-7B models trained with SFT+(LaF-GRPO) show leading performance, achieving a METEOR score of 0.542, substantially higher than GPT-4o. (3) Beyond quantitative gains, LaF-GRPO can potentially help generate people-centered instructions with enhanced linguistic variety, more intuitive directional cues, richer environmental details, and crucial safety considerations.

In summary, our main contributions are:

- We propose the LaF-GRPO framework, the first method to employ GRPO for NIG-VI with a novel LLM-simulated follower feedback;
- We contribute the NIG4VI benchmark, the first open-source comprehensive dataset featuring precise multi-modal navigation contexts to facilitate robust model evaluation for VI navigation;
- We present extensive empirical studies across VLMs under various paradigms (Zero-shot, Zero-(LaF-GRPO), SFT, and SFT+(LaF-GRPO)), demonstrating the effectiveness of our approach.

2 Related Work

2.1 VLMs and VIA

VLMs (Liu et al., 2023; Dai et al., 2023; OpenAI, 2024a; Anthropic, 2024; Team, 2024) have drawn attention for combining visual perception with language generation. Refining VLMs with Reinforcement Learning (Ouyang et al., 2022) improves alignment with human preferences and enhances reasoning abilities. Recent success in Group Relative Policy Optimization (GRPO) (DeepSeek-AI, 2025) has led to RL fine-tuned VLMs like VLM-R1 (Shen et al., 2025), AlphaDrive (Jiang et al., 2025), and MedVLM-R1 (Pan et al., 2025), broadening their application range. VIA with VLMs is closely related to visual captioning and Visual Question Answering (VQA). VIALM (Zhao et al., 2024) frames VIA as a VQA task, generating step-by-step guidance from environment images and user requests. While VIALM emphasizes environment-grounded guidance with tactile information, it is not specifically designed for navigation. WalkVLM (Yuan et al., 2025) extends this to dynamic walking assistance and introduces the Walking Awareness Dataset (WAD). Though WalkVLM tackles navigation, its focus remains on video captioning rather than precise orientation and mobility guidance.

2.2 Navigation Instruction Generation (NIG)

There are two main branches for NIG studies: NIG for embodied agents and NIG for the visually impaired. Prior research on NIG for embodied agents has predominantly focused on advanced visual processing techniques while generating trajectory-level instructions. More details can be found in Appendix A. For the NIG-VI branch, ASSISTER (Huang et al., 2022) introduced the UrbanWalk benchmark for in-situ instructional guidance and developed a navigation assistance model. Our work improves upon ASSISTER in two key ways: (1) we introduce a more detailed evaluation benchmark covering orientation, mobility, scene description, and safety warnings; and (2) we leverages advanced VLMs within a GRPO framework with an LLM-as-Follower reward mechanism, leading to more effective navigation instructions.

3 NIG-VI Task and LaF-GRPO Method

3.1 NIG-VI Task Formulation

We start this section by describing the NIG-VI task of generating in-situ step-by-step natural language instructions to guide VI users along a pre-planned route $P = [p_1, \ldots, p_K]$ using a VLM-based assistant system. The route P consists of positional waypoints p_i leading to a destination and is generated using the A* algorithm. At each discrete step i of the navigation, the VLM receives two primary inputs: a front-view camera image $x^{(i)}_{\text{image}}$ and a task description which includes the user's current pose $x^{(i)}_{\text{pose}} = (x^{(i)}_{\text{loc}}, x^{(i)}_{\text{rot}})$ represented by their

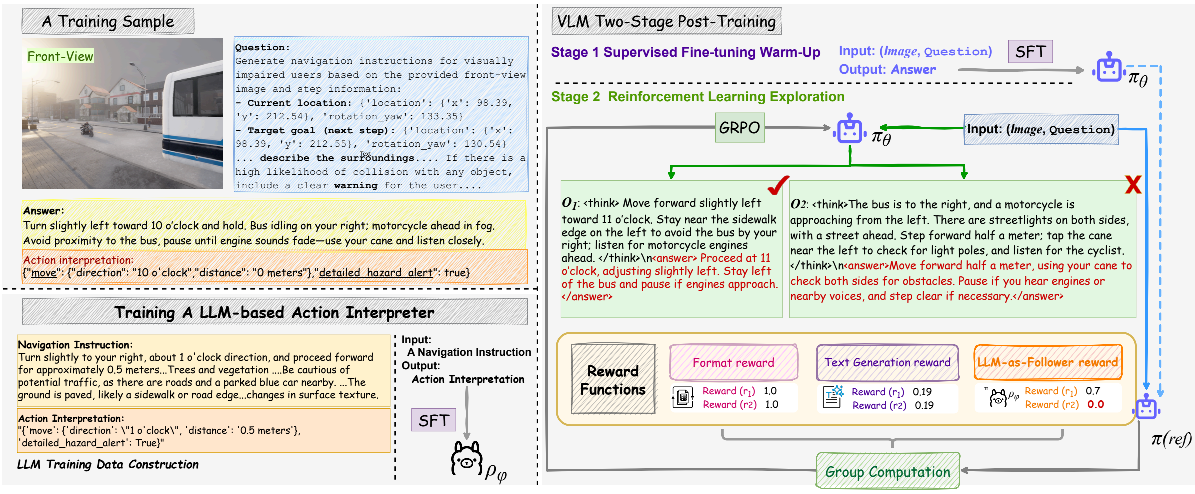


Figure 2: **Method Overview**. **Top left**: Training sample with input, target output, and generated navigation instruction's action interpretation. **Bottom left**: Action interpreter training using `LLaMA-3-8B-Instruct` to simulate VI users' navigation responses. **Right**: Post-training procedures for VLMs processing with LaF-GRPO using multiple reward functions (format, text generation, and LLM-as-Follower reward).

location $x_{\text{loc}}^{(i)} \in \mathbb{R}^3$ and rotation $x_{\text{rot}}^{(i)} \in \mathbb{R}^3$ within a global map coordinate system, as well as the next target waypoint $p_{i+1} \in P$. Based on these inputs, the VLM π generates a sequence of tokens $y = y^{(i)} = (y_1^{(i)}, y_2^{(i)}, \dots, y_t^{(i)})$ of token length t. The generated instruction y might also include details about the current surroundings captured in $x_{\text{image}}^{(i)}$ and any necessary safety alerts:

$$y_j \sim \pi_\theta(y_j^{(i)} | x_{\text{image}}^{(i)}, x_{\text{loc}}^{(i)}, x_{\text{rot}}^{(i)}, p_{i+1}, y_{<j}^{(i)}) \quad (1)$$

where θ denotes the adjustable model parameters.

3.2 The LaF-GRPO Framework

We then discuss our LaF-GRPO framework to tackle the NIG-VI task. It aims to address the challenges of *ensuring that navigation instructions are people-centered, practically usable by the VI users*, while *mitigating the need for costly real-world data collection with VI participants*. The overview of LaF-GRPO is illustrated in Figure 2, where the framework comprises two key components: (1) an LLM (without a visual encoder to "see") that simulates VI users' responses to navigation instructions by interpreting how these users would act upon hearing the instructions, and (2) a VLM post-training procedure that generates these instructions with (1)'s feedback. LaF-GRPO first employs an **action interpreter** to produce structured interpretations of potential user actions, which are then used to compute the LLM-as-Follower reward. This reward signal subsequently guides the VLM training process of **navigation instruction generation** that is more likely to be effectively followed by VI users in real-world navigation scenarios.

Action Interpreter. The action interpreter models VI user responses to navigation instructions. We fine-tune an LLM ρ with parameters φ to predict potential user actions from verbal guidance. Given VLM-generated instruction tokens y, it produces a structured action interpretation $\mathcal{A}$. Formally, we define $\mathcal{A}$ as a structured dictionary containing: (1) a *'move'* action with associated *'direction'* (indicated using clock positions) and *'distance'* parameters, and (2) a *'detailed_hazard_alert'* boolean flag that indicates whether the user perceives warnings about nearby obstacles, as illustrated in Figure 2 Left. To train such an action interpreter, we utilize training samples generated based on the prompt template detailed in Table 8 in Appendix E.

Navigation Instruction Generator. For VI guidance, we use a pre-trained VLM π with parameters θ for in-situ navigation instruction generation. The training of this generator involves two stages: Supervised Fine-tuning (SFT) and Group Relative Policy Optimization (GRPO). For SFT training details, please refer to Appendix B. For the GRPO, specifically, we propose LaF-GRPO, which is based on the standard GRPO (see below) reward function yet incorporates a novel LLM-as-Follower reward.

GRPO. The training process of GRPO aims to optimize the policy π_θ by maximizing the objective function $\mathcal{J}_{\text{GRPO}}(\theta)$. For a given query q, GRPO first samples a batch of G outputs $\{o_1, o_2, \dots, o_G\}$ using an older version of the policy, $\pi_{\theta_{\text{old}}}$. The training process of GRPO aims to optimize the policy π_θ by maximizing the objective function:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q,\{o_i\}\sim\pi_{\theta_{\text{old}}}}\left[\frac{1}{G}\sum_{i=1}^{G}\mathcal{L}_i - \beta\mathbb{D}_{\text{KL}}(\pi_\theta||\pi_{\text{ref}})\right] \tag{2}$$

Here, the term $\mathcal{L}_i$ represents the clipped surrogate objective used in PPO (Schulman et al., 2017):

$$\mathcal{L}_i = \min(w_i A_i, \text{clip}(w_i, 1-\epsilon, 1+\epsilon)A_i) \tag{3}$$

where $w_i = \frac{\pi_\theta(o_i|q)}{\pi_{\theta_{\text{old}}}(o_i|q)}$ is the importance sampling ratio, A_i is the estimated advantage for the output o_i, based on relative rewards of the outputs inside each group only, calculated as $A_i = \frac{r_i - \text{mean}(\{r_1,r_2,\ldots,r_G\})}{\text{std}(\{r_1,r_2,\ldots,r_G\})}$, and ϵ is a clipping hyperparameter. The second term, $-\beta\mathbb{D}_{\text{KL}}(\pi_\theta||\pi_{\text{ref}})$, regularizes the policy by penalizing divergence from a reference policy π_{ref} with coefficient β. This regularization stabilizes training by keeping the model close to the original effective policy, preventing it from losing previously learned capabilities.

3.3 LaF-GRPO Reward Functions

GRPO leverages verifiable rewards to simplify the reward modeling process. To integrate LLM's feedback to enable smooth, people-centered language guidance in NIG-VI, LaF-GRPO utilizes three reward functions as follows. The reward calculation algorithm is detailed in Algorithm 1 (Appendix C).

Format Reward. To encourage controllable generation for easy training, we adopt this binary reward ($r_{\text{format}} \in \{0, 1\}$) that evaluates structural compliance with the expected response format. Here, the reward would equal 1 if the output follows the required format pattern ‘<think>.*?</think>\n<answer>.*?</answer>’ in sequence, and 0 otherwise.

Text Generation Reward. We adopt METEOR to verify text generation rewards to align the generation to the ground-truth style. Here, METEOR is selected based on its evaluation of semantic overlap, incorporating synonymy and stemming to provide a nuanced, human-correlated assessment.

LLM-as-Follower Reward. To incorporate LLM's feedback for navigation, we propose the reward r_{LaF} to assess the navigational quality of generated instructions by comparing their interpreted actions (move direction, move distance, and alert flag) against those of a reference. Our intuition is that spatial factors, such as directional

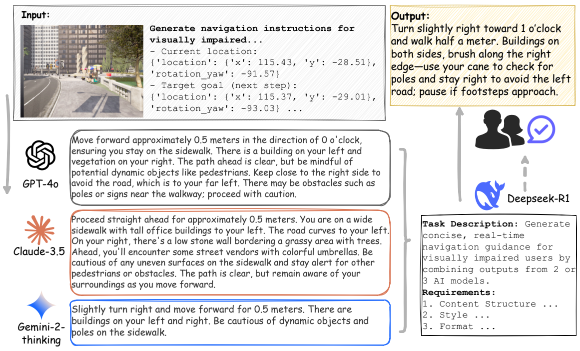


Figure 3: DeepSeek-R1 refines initial predictions from GPT-4o, Claude-3.5, and Gemini-2 into coherent instructions, using Vision-R1's modality bridging method (Huang et al., 2025). These are then reviewed and modified by human annotators for quality and accuracy.

accuracy (a_{dir}) and movement distance precision (a_{dist}), play a direct and critical role in determining navigation success. In contrast, safety alert flags (a_{alert}) serve as supplementary support for VI navigation by indicating potential hazards, though they are not primary determinants of success (Giudice and Legge, 2008; Younis et al., 2019). Considering these, the reward is computed as:

$$\begin{aligned} r_{LaF} = {} & w_{dir}\,\delta(a_{dir}, a_{dir}^{ref}) + w_{dist}\,\delta(a_{dist}, a_{dist}^{ref}) \\ & + w_{alert}\,\delta(a_{alert}, a_{alert}^{ref}) \end{aligned} \tag{4}$$

$\delta(\cdot)$ denotes exact match comparison. To prioritize spatial factors, weighting coefficients are set such that $w_{dir} + w_{dist} > w_{alert}$. This, in the end, yields an r_{LaF} score ranging from 0 to 1.

4 Benchmark: NIG4VI

We introduce the NIG4VI benchmark to address the scarcity of benchmark resources in this field. Inspired by the UrbanWalk, NIG4VI utilizes the open-sourced CARLA Simulator (Dosovitskiy et al., 2017) to collect samples from a diverse range of scenarios. These scenarios span remote rural-like settings (e.g., Town01) and complex metropolitan areas (e.g., Town10), and encompass various weather conditions, such as foggy and sunny weather. Pedestrian trajectories are generated using A* route planning algorithm, with precise geospatial coordinates, orientation, frontal-view images, and semantic segmentation images being recorded at each step. NIG4VI offers two main advantages: (1) its use of a realistic coordinate system facilitates easier transfer to real-world GPS applications, and (2) it enables the cost-effective generation of accurate and extensive data. Table 1 demonstrates NIG4VI's advantages compared to other datasets.

Benchmark	Level	# Samples	VIA	NIG	Spatial Acc.	Open-ended	Open-sourced
R2R (Anderson et al., 2018)	High	21k	✗	✓	✗	✓	✓
REVERIE (Qi et al., 2020)	High	10k / 6k	✗	✓	✗	✓	✓
UrbanWalk (Huang et al., 2022)	Detailed	2.6k	✓	✓	✓	✗	✗
Merchant et al. (2024)	Detailed	48	✓	✓	✗	✓	✗
VIALM (Zhao et al., 2024)	Detailed	200	✓	✗	✗	✓	✓
WAD (Yuan et al., 2025)	Detailed	12k / 120k	✓	✓	✗	✓	✓
NIG4VI (Ours)	Detailed	3k / 24k	✓	✓	✓	✓	✓
- w/o pre-calculation	Detailed	1.5k / 12k	✓	✓	✓	✓	✓
- with pre-calculation	Detailed	1.5k / 12k	✓	✓	✓	✓	✓

Table 1: Comparison of the NIG4VI dataset with existing benchmarks. Unlike WAD, NIG4VI employs geospatial coordinates for high spatial accuracy evaluation, and surpasses UrbanWalk through open-ended instruction generation that benchmarks VLMs' capacity to produce natural navigation guidance.

4.1 Dataset Construction

Each question's input includes the user's current location/rotation, the next step's location/orientation, and visual scene data. The complete prompt template used to structure these inputs is detailed in Table 6 of Appendix E. The synthesis of the output is a multi-stage process involving both advanced reasoning models and human annotation, as illustrated in Figure 3. Initially, several leading VLMs, specifically GPT-4o, Claude-3.5, and Gemini-2, generate predictions. Following a modality bridging approach, similar to that employed in Vision-R1 (Huang et al., 2025), these outputs are processed through DeepSeek-R1 to enhance blindness-oriented spatial guidance and navigability. The specific prompt guiding DeepSeek-R1 in this refinement task is detailed in Table 7.

Crucially, all instructions undergo rigorous human verification. This task is carried out by two annotators, both proficient in English and holding at least an undergraduate-level education, following a similar practice in (Zhao et al., 2024). The verification involves a two-stage process: first, one annotator performs initial content adjustments, adhering to task requirements. Subsequently, the second annotator reviews and verifies this work. Throughout this entire process, both annotators focus on ensuring: (1) elimination of visual references (e.g., color-based descriptors), (2) validation of non-visual landmarks, and (3) confirmation of metric precision for mobility-critical parameters.

4.2 Dataset Statistics

Table 2 details the statistics of the dataset, which comprises routes collected from six distinct towns within the CARLA simulator. Further details on route sampling in the CARLA simulator are in Appendix D. On average, each town contributes approximately 26.2 navigation routes. The average Euclidean distance between the start and end points of these routes is 113.51 units, with an average of 353.8 steps required for completion. After deduplication, the dataset yielded an average of 2,222.7 step-level (image, question) samples per town. It is partitioned into a training set of 1,500 samples from Town01 and a test set. The test set comprises the remaining 613 ***intra-town*** samples from Town01, along with all ***inter-town*** samples from Town02 (2,579), Town03 (2,260), Town04 (2,316), Town05 (1,935), and Town10 (2,133). Each data sample is available in two versions: *'with pre-calculation'* and *'without pre-calculation'*. The *'without pre-calculation'* version requires the VLM to independently calculate navigational parameters (e.g., distance, direction), presenting a greater challenge in guidance generation. Conversely, the *'with pre-calculation'* version provides the VLM with basic mathematical movement information. The VLM must then validate this data and assess the surroundings to generate the final navigation instruction.

Town	**Routes**	**Avg dist.**	**Avg steps**	**# Samples**
Town01	25	111.41	401	1,500 / 613
Town02	26	99.38	327	2,579
Town03	25	128.23	409	2,260
Town04	26	131.49	337	2,316
Town05	25	107.81	288	1,935
Town10	30	102.74	361	2,133
Avg.	**26.2**	**113.51**	**353.8**	**2,222.7**

Table 2: Statistics for sample routes. Dataset: 1,500 Town01 samples for training; the remaining 613 (Town01) and all other town samples for testing. *'Avg dist.'*: average Euclidean distance (route start to end). *'Avg steps'*: average steps per route. *'# Samples'*: deduplicated step-level (image, question) samples per town.

Pre-Cal.	Paradigm	Model	Intra-town ($N = 613$)				Inter-town (N =11,223)			
			BLEU ↑	ROUGE ↑	METEOR ↑	SPICE ↑	BLEU ↑	ROUGE ↑	METEOR ↑	SPICE ↑
No	Zero-Shot	DeepSeek-VL-7B	2.179	0.152	0.182	0.116	2.223	0.157	0.196	0.112
		MiniCPM-o-8B	2.009	0.145	0.234	0.131	1.969	0.142	0.233	0.129
		Intern-VL-8B	1.448	0.150	0.215	0.126	1.517	0.149	0.216	0.120
		Qwen-VL-7B	3.204	0.202	0.211	0.166	3.128	0.194	0.210	0.157
		GPT-4o	1.748	0.169	0.249	0.149	1.617	0.165	0.249	0.142
		Claude-3.5	2.803	0.216	0.304	0.211	2.749	0.211	0.301	0.202
		Gemin-2	4.105	0.236	0.232	0.232	4.422	0.252	0.238	0.236
	Zero-(LaF-GRPO)	Qwen-VL-3B	3.292	0.230	0.248	0.230	3.972	0.255	0.259	0.244
		Qwen-VL-7B	3.272	0.234	0.256	0.222	3.566	0.252	0.260	0.227
	SFT	Qwen-VL-3B	9.099	0.282	0.496	0.274	8.949	0.284	0.500	0.276
		Qwen-VL-7B	9.937	0.291	0.518	0.275	9.709	0.294	0.526	**0.281**
	SFT+(LaF-GRPO)	Qwen-VL-3B	**10.921**	**0.323**	0.528	0.274	**10.157**	**0.309**	0.527	0.276
		Qwen-VL-7B	10.037	0.284	**0.545**	**0.283**	9.002	0.276	**0.535**	0.278
Yes	Zero-Shot	DeepSeek-VL-7B	2.517	0.170	0.224	0.161	2.600	0.173	0.237	0.161
		MiniCPM-o-8B	2.349	0.166	0.210	0.136	2.517	0.177	0.220	0.144
		Intern-VL-8B	1.496	0.132	0.233	0.133	1.517	0.134	0.238	0.132
		Qwen-VL-7B	2.903	0.188	0.231	0.178	3.080	0.194	0.243	0.180
		GPT-4o	2.766	0.204	0.302	0.198	2.967	0.213	0.323	0.211
		Claude-3.5	4.124	0.236	0.349	0.257	3.400	0.214	0.326	0.224
		Gemin-2	5.132	0.252	0.266	0.269	6.144	0.276	0.283	0.284
	Zero-(LaF-GRPO)	Qwen-VL-3B	3.798	0.249	0.280	0.261	4.584	0.271	0.288	0.274
		Qwen-VL-7B	3.678	0.241	0.281	0.229	4.284	0.262	0.286	0.230
	SFT	Qwen-VL-3B	9.923	0.308	0.512	0.280	10.724	0.318	0.519	0.280
		Qwen-VL-7B	9.639	0.270	0.521	0.283	9.710	0.272	0.524	0.287
	SFT+(LaF-GRPO)	Qwen-VL-3B	**11.727**	**0.342**	0.541	0.286	**10.813**	**0.333**	0.535	0.279
		Qwen-VL-7B	10.499	0.285	**0.556**	**0.292**	9.232	0.275	**0.542**	**0.288**

Table 3: Evaluation results on the NIG4VI dataset across Intra-town and Inter-town subsets. Gray cells indicate results pertaining to Qwen2.5-VL models. Blue values highlight the best performing Qwen2.5-VL model within the Zero-Shot and Zero-(LaF-GRPO) categories. **Bold** values represent the highest score for each metric under a specific setting (with / without pre-calculation), while underlined values indicate the second-highest score.

5 Experimental Settings

Dataset. Experiments utilized the NIG4VI dataset, comprising Intra-town ($N = 613$) and Inter-town ($N = 11,223$) test subsets, under *'with/without pre-calculation'* conditions.

Models. Diverse VLMs were evaluated, falling into two main groups. The first group includes remote models: `GPT-4o` (OpenAI, 2024b), `Claude-3-5-sonnet-20240620` (Anthropic, 2024), and `Gemini-2.0-flash-thinking-exp-01-21` (Google DeepMind, 2024). The second group comprises smaller, locally runnable VLMs: `DeepSeek-VL-7B` (Lu et al., 2024), `MiniCPM-o-2.6-8B` (Yao et al., 2024), `Intern-VL-2.5-8B` (Chen et al., 2024), and `Qwen2.5-VL-3B/7B` (Bai et al., 2025).

Evaluation Metrics. Following previous studies in NIG (Huang et al., 2022; Fan et al., 2024; Kong et al., 2024), model performance was evaluated using a suite of widely adopted metrics: BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), METEOR (Banerjee and Lavie, 2005), and SPICE (Anderson et al., 2016). For each of these metrics, higher scores denote superior performance.

Baselines. We compare LaF-GRPO against two primary baseline methods: (1) **Zero-shot**: This involves models directly on NIG4VI without prior task-specific fine-tuning. (2) **Supervised Fine-tuning (SFT)**: Models are fine-tuned to generate instructions from the input. Furthermore, we implement two variants of LaF-GRPO to understand its different operational modes: (a) **Zero-(LaF-GRPO)**: LaF-GRPO is applied directly to the base model without SFT. (b) **SFT+(LaF-GRPO)**: LaF-GRPO is applied to models that have first undergone SFT. Both LaF-GRPO variants utilize the proposed LLM-as-Follower reward mechanism.

Implementation Deatails. LaF-GRPO training utilized a single NVIDIA H20 GPU (96 GB of memory). This hardware supports loading an 8B-param LLM (`LLaMA-3-8B`) and a 3B/7B-param Qwen2.5-VL model for LoRA (Hu et al., 2022)

Pre-Cal.	Reward Types			Intra-town (N = 613)				Inter-town (N = 11,223)			
	Format	**Meteor**	**LLM**	BLEU ↑	ROUGE ↑	METEOR ↑	SPICE ↑	BLEU ↑	ROUGE ↑	METEOR ↑	SPICE ↑
No	✓			10.251	0.318	0.524	0.278	9.401	0.304	0.523	0.279
	✓	✓		10.912	0.317	0.525	**0.279**	10.076	0.306	0.521	**0.279**
	✓	✓	✓	**10.921**	**0.323**	**0.528**	0.274	**10.157**	**0.309**	**0.527**	0.276
Yes	✓			11.269	0.337	0.538	**0.292**	10.217	0.328	0.530	**0.282**
	✓	✓		11.602	0.339	0.539	0.284	10.753	0.331	0.531	0.280
	✓	✓	✓	**11.727**	**0.342**	**0.541**	0.286	**10.813**	**0.333**	**0.535**	0.280

Table 4: Ablation study results for the Qwen2.5-VL-3B model on the NIG4VI dataset with different **reward functions**. Bold values represent the highest score for each metric under its specific pre-calculation condition.

Pre-Cal.	Model	Intra-town (N = 613)				Inter-town (N = 11,223)			
		BLEU ↑	ROUGE ↑	METEOR ↑	SPICE ↑	BLEU ↑	Rouge ↑	METEOR ↑	SPICE ↑
No	7B-format+meteor+LLM (**1k**)	9.401	0.283	0.529	0.274	8.963	0.281	0.530	0.275
	7B-format+meteor+LLM (**2k**)	9.657	0.280	0.539	0.276	9.001	**0.276**	**0.535**	0.274
	7B-format+meteor+LLM (**3k**)	**10.037**	**0.284**	**0.545**	**0.283**	**9.002**	**0.276**	**0.535**	**0.278**
Yes	7B-format+meteor+LLM (**1k**)	10.265	0.279	0.543	0.286	9.463	0.271	0.540	0.285
	7B-format+meteor+LLM (**2k**)	10.136	0.284	0.550	**0.292**	**9.245**	**0.276**	0.541	0.284
	7B-format+meteor+LLM (**3k**)	**10.499**	**0.285**	**0.556**	**0.292**	9.232	0.275	**0.542**	**0.288**

Table 5: Ablation study results for the Qwen2.5-VL-7B model on the NIG4VI dataset with varying training **sample sizes**. Bold values represent the highest score for each metric under its specific pre-calculation condition.

fine-tuning. The reward weights were configured as $(w_{dir}, w_{dist}, w_{alert}) = (0.4, 0.4, 0.2)$ based on analysis of navigation failure factors, prioritizing spatial parameters over contextual alerts. Training on $3k$ samples took approximately 15 hours, with the key hyperparameter group size G set to 8.

6 Results and Discussions

6.1 Main Results

We present Table 3, which summarizes model performance on NIG4VI, categorized by pre-calculation and training paradigms, and evaluated on intra-town and inter-town subsets. Comparing LaF-GRPO with the baselines reveals: (1) **Zero-Shot vs. Zero-(LaF-GRPO):** Zero-(LaF-GRPO) significantly enhances the Zero-Shot performance of VLMs, validating the effectiveness of LaF-GRPO. While the Zero-(LaF-GRPO) results suggest that increased model size (from 3B to 7B) does not necessarily guarantee improved performance across all metrics, it is noteworthy that for METEOR evaluations, specifically in intra-town scenarios, the 7B model achieved the highest scores (i.e., 0.256 and 0.281). This outcome may be attributable to the use of METEOR as a text generation reward during training and to the potentially more refined tuning applied to the 7B models. (2) **SFT & SFT+(LaF-GRPO):** SFT and SFT+(LaF-GRPO) yield significantly superior performance compared to Zero-Shot and Zero-(LaF-GRPO) models across all metrics and subsets, affirming the efficacy of fine-tuning. The SFT+(LaF-GRPO) approach further enhances performance beyond SFT. Moreover, under the SFT+(SFT-GRPO) paradigm, Qwen-VL-3B consistently achieves the highest BLEU and ROUGE scores, while Qwen-VL-7B excels in METEOR and SPICE. This performance pattern is observed for both intra-town and inter-town subsets and holds true regardless of pre-calculation. This may be attributable to 7B models demonstrating enhanced linguistic diversity in their outputs relative to 3B models. (3) **Additional Observations:** Scores are generally higher with pre-calculation than without, likely because it reduces the mathematical computation difficulty for the models. While both intra-town and inter-town results demonstrate consistent trends, intra-town evaluations typically yield higher scores. This is understandable as the intra-town test set comprises samples from the same town environments as the training set, leading to a closer data distribution.

6.2 Ablation Study

Reward Types Table 4 presents an ablation study investigating the impact of different reward types during SFT+(LaF-GRPO) training with the Qwen-VL-3B model. LaF-GRPO, incorporating the LLM-as-Follower reward, consistently achieves the highest BLEU, ROUGE, and METEOR scores. This trend holds true across both intra-town and inter-town evaluations, with or without pre-calculation.

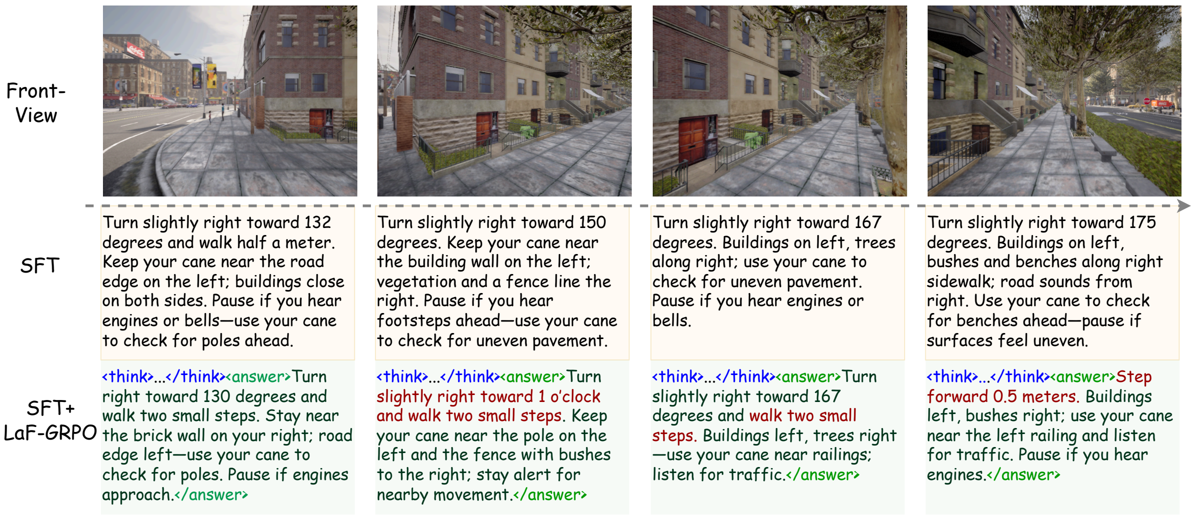


Figure 4: A comparative case study of navigational guidance provided by SFT and SFT+(LaF-GRPO) methods across successive steps. Findings indicate that SFT+(LaF-GRPO) (Ours) generates instructions with greater linguistic variety and more effectively incorporates o'clock directions and specific travel distances.

This underscores the significant benefit of the LLM-as-Follower reward signal for the NIG4VI task.

LaF-GRPO vs. Standard GRPO We conducted an additional experiment on the inter-town subset. It reveals that LaF-GRPO-trained models demonstrate superior navigational accuracy (68.1% vs. 67.3%) and their instructions were more frequently selected by GPT-4o for helpfulness and clarity (58.3% vs. 41.7% of cases). We identify two primary advantages of LaF-GRPO over standard GRPO, which utilizes only format and text generation rewards: (1) **Navigational Accuracy:** LaF-GRPO provides more precise movement and orientation accuracy. (2) **Instruction Clarity:** The inclusion of an action interpreter requirement encourages VLMs to produce instructions comprehensible to the follower and thus clearer and more structured. Details are provided in Appendix F.

Training Sample Sizes Table 5 presents an ablation study on the Qwen2.5-VL-7B model trained with SFT+(LaF-GRPO), illustrating the effect of varying training sample sizes ($1k$, $2k$, and $3k$). For the two metrics METEOR and SPICE, which are often considered more comprehensive in text generation, increasing the volume of training data generally leads to enhanced performance. Across the majority of evaluated conditions (with/without pre-calculation), scaling up to $3k$ samples typically yields the optimal or near-optimal scores. Nevertheless, for the 7B models in the inter-town setting, training with $2k$ samples also achieves comparable METEOR scores (i.e., 0.535 and 0.541), indicating training data efficiency at this sample size.

6.3 Case Study

Figure 4 provides a qualitative comparison of our SFT+(LaF-GRPO) method against the SFT baseline. Notably, SFT+(LaF-GRPO) generates instructions with greater linguistic variety and more intuitive directional cues. For instance, in Step 2, SFT+(LaF-GRPO) employs an o'clock direction (*"Turn slightly right toward 1 o'clock"*) and a relatable distance (*"two small steps"*), contrasting with SFT's numerical bearing (*"150 degrees"*). This approach can yield guidance that is more naturally understood by VI users. Furthermore, SFT+(LaF-GRPO), leveraging its internal reasoning process (i.e. the <think>...</think> blocks), frequently incorporates more environmental details and safety considerations. For example, its instruction for Step 4 (*"Step forward 0.5 meters; ...use your cane near the left railing and listen for traffic"*) also emphasizes immediate safety interactions. Full instruction texts for Zero-(LaF-GRPO), SFT, and SFT+(LaF-GRPO) are available in Appendix F.

7 Conclusion

This study addresses navigation instruction generation for the visually impaired individuals. We constructed the NIG4VI benchmark. Following this, we developed LaF-GRPO, a training paradigm for VLMs that incorporates an LLM-as-Follower reward. Experimental evaluations established LaF-GRPO's superiority over baselines and standard GRPO, with future qualitative analysis confirming the generated instructions' real-world practicality.

Limitations

The main limitations of this work include two primary aspects: the data source for the benchmark NIG4VI and the computational demands of our proposed training methodology, LaF-GRPO. (1) The first limitation pertains to the benchmark NIG4VI, as its data is collected within a simulated environment. A domain gap inevitably exists between such simulated conditions and the multifaceted complexities of the real world, as simulators may not fully capture the entire spectrum of real-world environmental dynamics. Nevertheless, simulators facilitate large-scale data acquisition, enable precise and automated annotation, and provide a crucial foundation for initial model development and systematic evaluation. Future work will focus on incorporating more diverse, real-world data and exploring sim-to-real transfer techniques to mitigate this gap. (2) Second, training the LaF-GRPO model is resource-intensive, requiring substantial computational power and considerable time. However, this high cost is primarily a one-time investment and does not significantly affect deployment. Once trained, the LaF-GRPO model operates efficiently during inference, ensuring practical, responsive real-time navigation instructions for end-users.

Ethical Considerations

Our work has been conducted with a strong commitment to ethical practices and transparency throughout the development process: (1) **Licensing**: Data collection was performed using the CARLA Simulator, which is distributed under an MIT license. Our use of CARLA aligns with its original intended purpose and adheres to its licensing terms. Our research incorporates publicly available open-source VLMs, specifically the Qwen2.5-VL-3B and Qwen2.5-VL-7B models. These models are licensed under Apache 2.0. This permissive licensing allows code modification, fostering innovation and broader application. (2) **Instruction Biases and Safety**: Our method generates navigation instructions. A core design principle of our NIG-VI system is safety awareness. This focus on safety and bias mitigation supports the responsible application of AI in navigation technologies.

References

Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. *Preprint*, arXiv:1607.08822.

Peter Anderson, Qi Wu, and et al. 2018. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, pages 3674–3683. Computer Vision Foundation / IEEE Computer Society.

Anthropic. 2024. The claude 3 model family: Opus, sonnet, haiku.

Anthropic. 2024. Claude 3.5 sonnet.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo 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, and 8 others. 2025. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Zhe Chen, Jiannan Wu, and et al. 2024. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198.

Wenliang Dai, Junnan Li, and et al. 2023. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *NeurIPS*.

DeepSeek-AI. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16.

Sheng Fan, Rui Liu, Wenguan Wang, and Yi Yang. 2024. Navigation instruction generation with BEV perception and large language models. In *ECCV*, volume 15080 of *Lecture Notes in Computer Science*, pages 368–387. Springer.

Nicholas A Giudice and Gordon E Legge. 2008. Blind navigation and the role of technology. *The engineering handbook of smart technology for aging, disability, and independence*, pages 479–500.

Google DeepMind. 2024. Introducing gemini 2.0: Our new ai model for the agentic era. https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024. Accessed: 2025-01-21. Published: December 2024.

Muraleekrishna Gopinathan, Martin Masek, Jumana Abu-Khalaf, and David Suter. 2024. Spatially-aware speaker for vision-and-language navigation instruction generation. In *ACL*, pages 13601–13614. Association for Computational Linguistics.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Wenxuan Huang, Bohan Jia, and et al. 2025. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *Preprint*, arXiv:2503.06749.

Zanming Huang, Zhongkai Shangguan, Jimuyang Zhang, Gilad Bar, Matthew Boyd, and Eshed Ohn-Bar. 2022. ASSISTER: assistive navigation via conditional instruction generation. In *ECCV*, volume 13696 of *Lecture Notes in Computer Science*, pages 271–289. Springer.

Bo Jiang, Shaoyu Chen, Qian Zhang, Wenyu Liu, and Xinggang Wang. 2025. Alphadrive: Unleashing the power of vlms in autonomous driving via reinforcement learning and reasoning. *Preprint*, arXiv:2503.07608.

Xianghao Kong, Jinyu Chen, Wenguan Wang, Hang Su, Xiaolin Hu, Yi Yang, and Si Liu. 2024. Controllable navigation instruction generation with chain of thought prompting. In *ECCV*, volume 15087 of *Lecture Notes in Computer Science*, pages 37–54. Springer.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *NeurIPS*.

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Yaofeng Sun, Chengqi Deng, Hanwei Xu, Zhenda Xie, and Chong Ruan. 2024. Deepseek-vl: Towards real-world vision-language understanding. *Preprint*, arXiv:2403.05525.

Zain Merchant, Abrar Anwar, Emily Wang, Souti Chattopadhyay, and Jesse Thomason. 2024. Generating contextually-relevant navigation instructions for blind and low vision people. *Preprint*, arXiv:2407.08219.

OpenAI. 2024a. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

OpenAI. 2024b. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Long Ouyang, Jeffrey Wu, and et al. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Jiazhen Pan, Che Liu, Junde Wu, Fenglin Liu, Jiayuan Zhu, Hongwei Bran Li, Chen Chen, Cheng Ouyang, and Daniel Rueckert. 2025. Medvlm-r1: Incentivizing medical reasoning capability of vision-language models (vlms) via reinforcement learning. *Preprint*, arXiv:2502.19634.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318. ACL.

Yuankai Qi, Qi Wu, and et al. 2020. REVERIE: remote embodied visual referring expression in real indoor environments. In *CVPR*, pages 9979–9988. Computer Vision Foundation / IEEE.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, Ruochen Xu, and Tiancheng Zhao. 2025. Vlm-r1: A stable and generalizable r1-style large vision-language model. *Preprint*, arXiv:2504.07615.

Gemini Team. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, and 1 others. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.

Ola Younis, Waleed Al-Nuaimy, Fiona Rowe, and Mohammad H Alomari. 2019. A smart context-aware hazard attention system to help people with peripheral vision loss. *Sensors*, 19(7):1630.

Zhiqiang Yuan, Ting Zhang, Ying Deng, Jiapei Zhang, Yeshuang Zhu, Zexi Jia, Jie Zhou, and Jinchao Zhang. 2025. Walkvlm:aid visually impaired people walking by vision language model. *Preprint*, arXiv:2412.20903.

Yi Zhao, Yilin Zhang, Rong Xiang, Jing Li, and Hillming Li. 2024. Vialm: A survey and benchmark of visually impaired assistance with large models. *Preprint*, arXiv:2402.01735.

Appendix

A Related Work Details

General NIG for Embodied Agents Prior research on navigation instruction generation (NIG) for embodied agents has predominantly focused on advanced visual processing techniques while generating trajectory-level instructions. BEVInstructor (Fan et al., 2024) employs a Bird's-Eye View encoder and iterative refinement for clearer instructions. SAS (Gopinathan et al., 2024) uses structural and semantic knowledge with adversarial reward learning to improve instruction quality. C-Instructor (Kong et al., 2024) focuses on style-controlled instruction generation and adopts a chain-of-thought with landmarks mechanism. Our approach differs in two significant ways: (1) while existing methods emphasize visual representation techniques, our LaF-GRPO approach prioritizes navigation feedback for VLM fine-tuning; and (2) traditional NIG systems generate trajectory-level instructions for complete routes, whereas NIG-VI scenario provides step-level in-situ instructions.

B SFT Details

SFT The objective of SFT is to maximize the likelihood of the generated instruction y given the input image and question. The output $y = \{y_1, y_2, ..., y_t\}$ is a sequence of navigation instruction tokens. The input question x_{question} includes positional information x_{loc}, x_{rot}, and p_{i+1}. The loss function is defined as:

$$\mathcal{L}_{\text{SFT}} = -\sum_{t=1}^{T} \log P_\theta(y_t \mid y_{<t}, x_{\text{image}}, x_{\text{question}}), \tag{5}$$

where where y_t represents the t-th token of the navigation instruction, $y_{<t}$ denotes all preceding tokens, and T is the total instruction length.

C Algorithm Details

D Route Sampling in CARLA

Figure 5 illustrates BEV maps of CARLA towns with sampled routes highlighted in blue. Figure 6 illustrates examples of sampled start and end points for pedestrians in CARLA Town03 and Town10. Along a given route or trajectory, front-facing RGB images and corresponding semantic segmentation views can be collected (Figure 7).

Algorithm 1 LaF-GRPO Reward Calculation

Require: Generated output o_i, Reference output o_{ref}, Weights α, β, γ (default)
Ensure: Combined reward score r_i

function CALCULATEREWARD(o_i, o_{ref})
 ▷ *Format Reward*: $r^{(1)} \leftarrow 1/0$
 ▷ *Text Generation Reward*:
 $r^{(2)} \leftarrow \text{METEOR}(o_i, o_{\text{ref}})$
 ▷ ***LLM-as-Follower Reward***:
 $a_i \leftarrow \text{ExtractAction}(o_i)$
 $a_{\text{ref}} \leftarrow \text{ExtractAction}(o_{\text{ref}})$
 $r^{(3)} \leftarrow \text{action\_compare}(a_i, a_{\text{ref}})$
 ▷ Compute weighted reward:
 $r_i \leftarrow \alpha \cdot r^{(1)} + \beta \cdot r^{(2)} + \gamma \cdot r^{(3)}$
 return r_i
end function

E Prompt Details

Table 6 presents a detailed prompt template. This specific version is formulated without pre-calculation. For experimental conditions employing a "with pre-calculation" approach, this base prompt is augmented by an additional sentence that explicitly states the calculated distance and direction, such as: "The movement direction is [DIRECTION] direction with a distance of [DISTANCE]." Table 7 outlines the prompt used for generating step-level instructions with the DeepSeek-R1 model. Finally, Table 8 describes the prompt for generating action interpreter samples.

F Results Details

Additional Observations Scores are generally higher with pre-calculation than without. This is likely because pre-calculation reduces the mathematical computation difficulty for the models. Furthermore, while both intra-town and inter-town results demonstrate consistent trends, intra-town evaluations typically yield higher scores. This is understandable as the intra-town test set comprises samples from the same town environments as the training set, leading to a closer data distribution.

LaF-GRPO vs. Standard GRPO Table 9 describes the prompt used by GPT-4o to calculate navigational accuracy, while Table 10 presents the prompt for GPT-4o to select a better navigation instruction. Examples are in Table 11 and Table 12.

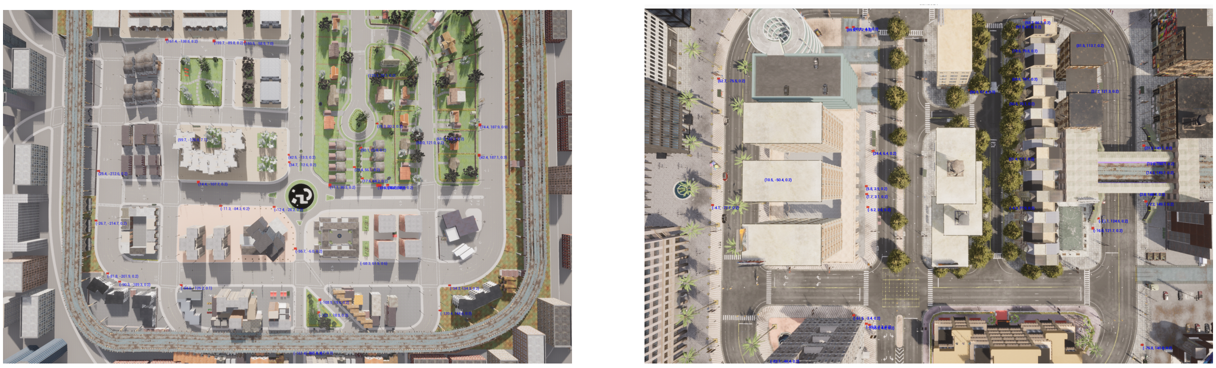
Case Study Figure 8 compares the navigational guidance outputs and instructions generated by Zero-(LaF-GRPO), SFT, and SFT+(LaF-GRPO).

(a) Town01 (b) Town02

(c) Town03 (d) Town04

(e) Town05 (f) Town10

Figure 5: BEV maps of CARLA towns with sampled routes highlighted in blue



(a) Town03 (b) Town10

Figure 6: Examples of sampled start and end points for pedestrians in CARLA Town03 and Town10.

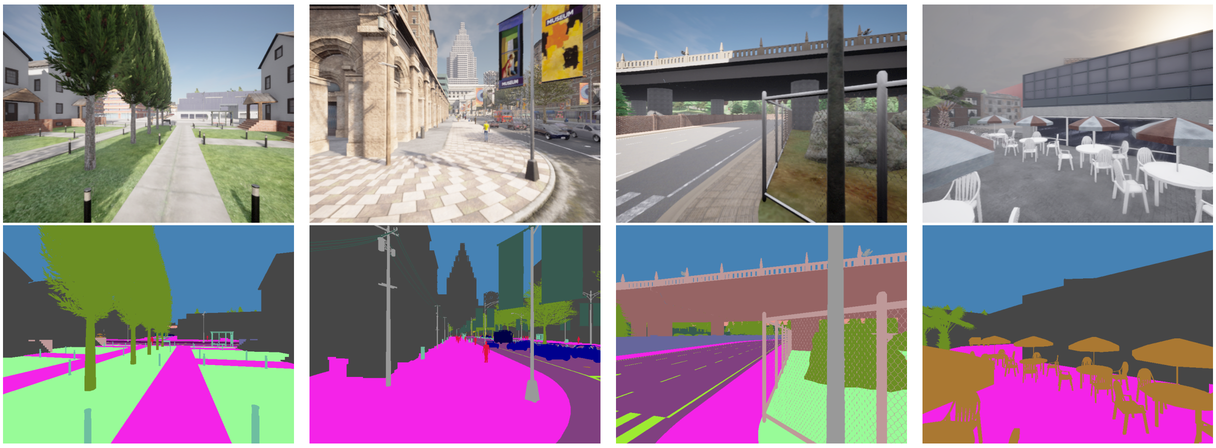


Figure 7: Example pairs of front-facing RGB images and their corresponding semantic segmentation views.

Prompt Template

Generate navigation instructions for visually impaired users based on the provided front-view image and step information:
- Current location:
{*'location'*: {*'x'*: [START_X], *'y'*: [START_Y]},
'rotation_yaw': [START_YAW]}
- Target goal (next step):
{*'location'*: {*'x'*: [TARGET_X], *'y'*: [TARGET_Y]}, *'rotation_yaw'*: [TARGET_YAW]}
Determine the direction and distance to the next step:
- If the angle is between -15° and 15°, the direction is straight ahead ("0 o'clock").
- If the angle is between 15° and 45°, the direction is slightly to the right ("1 o'clock").
- For angles beyond this, map the directions incrementally in 30° steps (e.g., "2 o'clock" through "11 o'clock").
- For angles near ±180° or wraparound (165° to -165°), the direction is directly behind ("6 o'clock").
- Calculate the distance using the Euclidean distance formula.
Semantic view information:[
'*top_left*': [TOP_LEFT_SEM_OBJ],
'*top_middle*': [TOP_MID_SEM_OBJ],
'*top_right*': [TOP_RIGHT_SEM_OBJ],
'*bottom_left*': [BOTTOM_LEFT_SEM_OBJ],
'*bottom_middle*':[BOTTOM_MID_SEM_OBJ],
'*bottom_right*':[BOTTOM_RIGHT_SEM_OBJ]]
- Incorporate this information to describe the surroundings in combination with the image. If there is a high likelihood of collision with any object, include a clear warning for the user.
Provide only the final navigation instructions for the step, including spatial and environmental details, as well as potential safety warnings. Do not include any additional or unrelated text.

Table 6: Detailed prompt template for navigation instructions for visually impaired users (without Pre-Calculation).

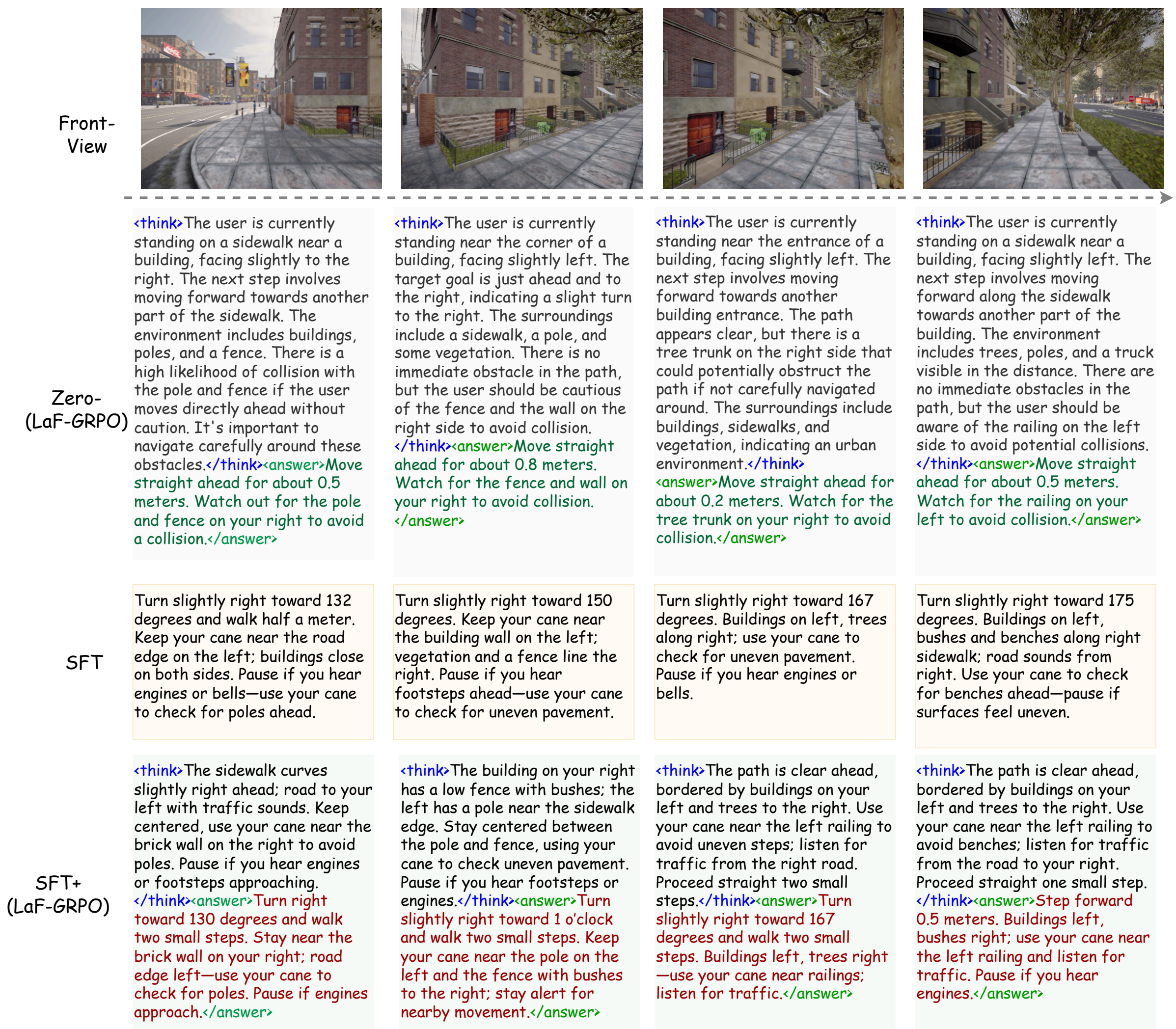


Figure 8: Case study of Zero-(LaF-GRPO), SFT and SFT+(LaF-GRPO) navigational guidance over steps.

Prompt for Generating Step-Level Instructions with DeepSeek-R1

***Task Description**: Generate concise, real-time navigation guidance for visually impaired users by combining outputs from 2 or 3 AI models.*

***Requirements**:*

***1. Content Structure must include**: (1) Action: Provide clock-direction references (e.g., "11 o'clock") with metric distances (e.g., "0.5 meters") or relatable analogies (e.g., "two small steps"). (2) Environment: Describe left, right, and ahead in 5-10 words, focusing on tactile or sound cues (e.g., "bushes on left", "traffic sounds from right"). (3) Hazards: Explicitly identify immediate dangers (e.g., "benches", "road edge") with actionable warnings (e.g., "pause and listen for bike bells").*

***2. Style**: Use natural, conversational language, avoiding robotic terms. Prioritize critical information, removing redundant descriptions, and incorporate sensory guidance when necessary (e.g., "use your cane to check", "listen for engines").And avoid words like "watch" or "see" since visually impaired individuals may not be able to perceive their environment through sight.*

***3.Format**: Limit to 2-3 sentences. Output only the polished instruction text.*

Output the concise navigation guidance text only.

*This is **an output example**: Turn slightly left toward 11 o'clock and walk half a meter. Keep your cane near the bushes on the left to avoid benches; listen closely for traffic from the right road. Pause if you hear engines or bells.*

The model 1's generated text is: [GPT-4o's INSTRUCTION]
The model 2's generated text is: [Gemini-2's INSTRUCTION]
The model 3's generated text is: [Claude-3.5's INSTRUCTION]

You must make sure output the polished instruction only, without additional words!
Now output the polished result:

Table 7: Prompt for Generating Step-Level Instructions with DeepSeek-R1

Prompt for Generating Action Interpreter Samples with DeepSeek-R1

Assume you are a blind person. Analyze the instruction through these steps:
1. Determine if movement parameters exist:
- Extract direction: Convert any directional information (e.g., left, right, east, west) into the "X o'clock" format (e.g., "2 o'clock" for a slight right turn, "9 o'clock" for a left turn).
- Extract distance: Ensure the distance includes a numerical value and a unit (e.g., meters, steps).
2. Check for danger alerts:
- Identify if the instruction includes detailed warnings about hazards (e.g., specific obstacles at specific directions or distances). If hazards are mentioned but lack detail, consider the alert as non-detailed.
3. If both direction and distance are missing, or if the instruction is unclear or ambiguous, return None for the movement parameters

Output Format:
```
{
    "move": {
        "direction": "X o'clock", // Replace "X" with the appropriate value (e.g., "2 o'clock").
        "distance": "Y meters/steps" // Replace "Y" with the numerical value and unit.
    },
    "detailed_hazard_alert": true/false // Set to ‘true‘ if detailed hazard warnings are present, otherwise ‘false‘.
}
```
or, if the instruction is invalid or incomplete:
```
{
    "move": None,
    "detailed_hazard_alert": true/false // Set to ‘true‘ if any hazard warnings are present, even if incomplete.
}
```

Examples:
Example 1:
Input: ‘Walk forward approximately 0.5 meters, maintaining your current direction (0 o'clock). The surroundings include a sidewalk on the right with walls and vegetation, buildings to the left, and roads ahead.'
Output: {"move": {"direction": "0 o'clock", "distance": "0.5 meters"},"detailed_hazard_alert": false}
Example 2:
Input: ‘Turn left, and move forward to avoid the building on your right.'
Output: {"move": {"direction": "9 o'clock", "distance": "0 meters"}, "detailed_hazard_alert": true}
Example 3:
Input: ‘Caution: wet floor ahead.'
Output: {"move": None,"detailed_hazard_alert": true}

Now, solve the following task:
Input Instruction: [INPUT INSTRUCTION]
Output:

Table 8: Prompt for Generating Action Interpreter Samples with DeepSeek-R1

Prompt Template for Navigational Accuracy Calculation

I will provide you with one prediction and one reference_ground_truth. Your task is to evaluate the prediction against the reference_ground_truth based on two criteria: movement direction and distance. You will then return a score of 1, 0.5, or 0 based on the following rules:

*Return **1** if:*

– The movement direction in the prediction is correct when compared to the reference_ground_truth, AND

– The distance in the *prediction* is correct when compared to the *reference_ground_truth*.

*Return **0.5** if:*

– The movement direction is correct BUT the distance is incorrect, OR

– The movement direction is incorrect BUT the distance is correct.

*Return **0** if:*

– The movement direction in the prediction is incorrect when compared to the reference_ground_truth, AND

– The distance in the prediction is incorrect when compared to the reference_ground_truth.

Prediction: [PREDICTION]

Reference Ground Truth: [REFERENCE]

Output only the score without additional words.

Table 9: Prompt Template for Navigational Accuracy Calculation.

Prompt template for Instruction Clarity Evaluation

I will provide you with two predictions: pred_1 and pred_2. Your task is to determine which of these two predictions offers clearer and better-structured instructions.

Evaluate pred_1 and pred_2 based on the following criteria for instruction quality:

***Movement Guidance:** How clear, concise, and actionable is the guidance for movement?*

***Environment Surrounding:** How well are relevant details about the surrounding environment described?*

***Potential Safety Alerts:** Is necessary information regarding potential safety hazards included appropriately?*

Based on your comparative evaluation of these criteria, decide which prediction is better.

pred_1: [PREDICTION_1]

pred_2: [PREDICTION_2]

Output: Return "1" if pred_1 is better, or "2" if pred_2 is better.

Output only 1 or 2 without additional words.

Table 10: Prompt Template for Evaluating Instruction Clarity.

Types	Info	Format+METEOR	Format+METEOR+LLM (Ours)
Navigational Accuracy	Town05, w/o pre-calculation move_gt: (11 o'clock, 0.5 meters)	**<think>** *Move forward half a meter toward 12 o'clock. Buildings line your left; road sounds from the right. Avoid poles ahead—use your cane to check pavement.* **</think>** **<answer>** *Turn slightly left toward 11 o'clock and walk half a meter. Keep your cane near the buildings on the left; pause if you hear traffic approaching from the right.* **</answer>**	**<think>** *Turn slightly left toward 11 o'clock and walk half a meter. Keep your cane near the building edge on the right to avoid poles; traffic sounds from the left road. Pause if you hear engines or bells.* **</think>** **<answer>** *Turn slightly left toward 11 o'clock and walk half a meter. Buildings on your right, road sounds from the left; use your cane to check for poles and pause if engines approach.* **</answer>**
	Town05, w/o pre-calculation move_gt: (10 o'clock, 0.5 meters)	**<think>** *Move forward half a meter toward 12 o'clock. Keep your cane near the bushes on the left; road edge and rocks to the right. Pause if you hear traffic and check for poles with your cane.* **</think>** **<answer>** *Proceed straight at 12 o'clock for half a meter. Use your cane to avoid poles and the rocky wall on the right; listen closely for engines.* **</answer>**	**<think>** *The road curves left toward 10 o'clock, with bushes on your left and a rocky cliff to the right. A pole is ahead at 12 o'clock; steer slightly left to avoid it and check the sidewalk edge with your cane.* **</think>** **<answer>** *Turn slightly left toward 11 o'clock and take two small steps. Use your cane to check the left curb and avoid the pole ahead; listen for traffic from the right road.* </answer>
	Town10, with pre-calculation move_gt: (11 o'clock, 0.2 meters)	**<think>***Move slightly right toward 1 o'clock for 0.2 meters. Keep your cane near the building edge on the left; traffic sounds from the right road. Pause at the pole ahead—check for uneven pavement with your cane.* **</think>** **<answer>** *Turn slightly right toward 1 o'clock and walk 0.2 meters. Use your cane to check the sidewalk edge and pause if you hear traffic approaching.* **</answer>**	**<think>***Turn slightly left toward 11 o'clock and walk 0.2 meters. Keep your cane near the building on the right to avoid the traffic pole; listen for engines from the road to your left.* **</think>** **<answer>** *Turn slightly left toward 11 o'clock and walk 0.2 meters. Building on your right, road to the left; use your cane to check for the traffic pole ahead and pause if you hear engines.* **</answer>**

Table 11: Example comparison of different formats for Navigational Accuracy

Types	Info	Format+METEOR	Format+METEOR+LLM (Ours)
Instruction Clarity	Town05, w/o pre-calculation move_gt: (12 o'clock, 0.5 meters)	**<think>** *Move forward half a meter toward 12 o'clock. Buildings line your left; road sounds from the right. Avoid potted plants ahead—pause if engines approach.* **</think>** **<answer>** *Proceed straight at 12 o'clock for half a meter. Use your cane to check for planters and listen for traffic.* **</answer>**	**<think>** *Move forward half a meter toward 12 o'clock. Buildings line your left; road sounds from the right. Avoid potted plants ahead—use your cane to check.* **</think>** **<answer>** *Proceed straight at 12 o'clock for half a meter. Buildings on left, road sounds right; use your cane to navigate around potted plants ahead. Pause if engines approach.* **</answer>**
	Town03, with pre-calculation move_gt: (12 o'clock, 0.5 meters)	**<think>** *Move slightly left toward 11 o'clock for half a meter. Keep your cane near the bushes on the left to avoid the bench; listen for traffic from the right road. Pause if you hear engines or bells.* **</think>** **<answer>** *Turn left toward 11 o'clock and walk half a meter. Use your cane to check the left hedge; pause for traffic sounds from the right road.* **</answer>**	**<think>** *Move slightly left toward 11 o'clock for half a meter. Keep your cane near the bushes on the left to avoid the bench; listen for traffic from the right road.* **</think>** **<answer>** *Turn left toward 11 o'clock and walk half a meter. Bushes on your left, road sounds from the right; use your cane to check the path and pause if you hear engines.* **</answer>**
	Town10, with pre-calculation move_gt: (1 o'clock, 0 meters)	**<think>** *Turn slightly right toward 1 o'clock and take two small steps. Keep your cane near the building on the left; traffic sounds from the right road. Pause if you hear engines or bells.* **</think>** **<answer>** *Turn slightly right toward 1 o'clock and take two small steps. Use your cane near the building on the left; pause for traffic sounds from the right.* **</answer>**	**<think>** *Turn slightly right toward 1 o'clock and take two small steps. Keep your cane near the building on the left; traffic sounds from the right road. Pause if engines approach and check for uneven pavement.* **</think>** **<answer>** *Turn slightly right toward 1 o'clock and take two small steps. Buildings on your left, road with trucks to the right; use your cane to avoid poles and pause if engines approach.* **</answer>**

Table 12: Example comparison of different formats for Instruction Clarity