GA-VLN: Geometry-Aware BEV Representation for Efficient Vision-Language Navigation

Supplementary Material

A. Real-World Robot Experiments

To validate the effectiveness and generalizability of the proposed GA-VLN framework, we deploy it onto the physical robot and conduct qualitative VLN experiments in real-world environments.

Setup. We use the Hello Robot Stretch 3 platform, a wheeled mobile system capable of executing low-level actions (e.g., moving forward and turning left/right) while providing RGB-D observations and odometry feedback. The agent's observations are transmitted via Wi-Fi to a local workstation for inference, after which the agent executes the predicted actions. The experiments are performed in a real apartment of approximately $60\,\mathrm{m}^2$, closely matching the layout of the simulator environments. To mitigate the domain gap between simulation and reality, we introduced specific adaptations for stable deployment: (1) the rotational step angle was adjusted to 15°; (2) the RGB-D frame is captured at every step and the BEV representation aggregated up to 16 historical frames to handle real-world sensory noise; and (3) the two-round dialogue format described in Sec. 3.3 was disabled to streamline inference, the BEV representation is updated every four steps. All other settings remain identical to those used in the simulator-based experiments, and no additional modules (e.g., obstacle avoidance or navigable-point filtering) are introduced.

Results. As shown in Fig. 4 and 5, GA-VLN enables the agent to navigate accurately in real-world environments based on natural-language instructions, including both complex fine-grained descriptions (example #1 and #3) and high-level goal-directed commands (example #2). The figures illustrate the navigation process from both third-person and first-person views, along with a visualization of the projection of visual patch features into the BEV space during execution. Especially from the first-person view, the navigation trajectories appear reasonable and coherent, and in demonstrated cases the agent successfully reaches the described targets (e.g., the bed, the television, or the blue door). For ease of demonstration, the BEV visualization is shown in an absolute coordinate frame; in practice, GA-BEV operates by projecting features into an ego-centric BEV coordinate system. These visualizations reveal that the agent maintains a meaningful geometric understanding of the environment—for example, it roughly covers the regions it has observed and is able to delineate major structural contours such as walls. Together, these qualitative results demonstrate the effectiveness of GA-VLN in real-world deployment.

Limitations. We observed distinct challenges in zero-shot real-world transfer. Without auxiliary obstacle avoidance modules, the agent occasionally executed paths dangerously close to obstacles (e.g., hugging walls), as it optimized for the shortest path trained in simulation. Additionally, the coarse granularity of discrete actions sometimes led to imprecise stopping behavior. These observations highlight the limitations of directly deploying the model in unmodified real-world environments. Nevertheless, despite operating without any auxiliary modules, GA-VLN shows strong instruction comprehension and produces reliable action sequences, indicating its potential as a foundational model for real-world navigation.

B. More details of Geometry-Aware VLN Framework

By integrating the proposed GA-BEV representation, we develop an efficient Vision-Language Navigation (VLN) framework that enables spatially grounded reasoning with compact visual tokens for navigation.

Specifically, at each navigation step t, given the language instruction L, the current-view visual features V_t, and the geometry-aware BEV features B aggregated from V_t and up to the last eight front-view observations $\{V_{t-1}, \ldots, V_{t-8}\}$, the MLLM predicts an action sequence A_t consisting of four discrete actions from the action vocabulary $\mathcal{A} = \{\uparrow, \leftarrow, \rightarrow, \texttt{STOP}\}$:

$$A_t = f_{\text{MLLM}}(L, B, V_t) \tag{6}$$

After the agent completes the four actions in A_t, we denote its current front-view feature as V_{t+1}, and then proceed to predict A_{t+1}. Besides, we adopt a two-round dialogue format for action prediction, following [39]. Specifically, after the agent finishes the four actions in A_t, we do not immediately update the BEV feature B. Instead, the next prediction is made based only on the current V_{t+1}, A_t, and the input used in the previous prediction:

$$A_{t+1} = f_{\text{MLLM}}(L, B, V_t, A_t, V_{t+1}) \tag{7}$$

After executing the four actions in A_{t+1}, the two-round dialogue process, Eq. 6 and Eq. 7, is completed and a new

dialogue begins. At this point, the current front-view feature becomes V_{t+2}. We then construct a new BEV feature using V_{t+2} and up to eight previous frames $\{V_{t+1}, V_t, \ldots, V_{t-6}\}$, and predict A_{t+2} following Eq. 6. Therefore, the BEV feature is updated once every 8 actions.

During training, we divide each full trajectory into examples, each consisting of 8 actions, and train the model using the prompt format described above. During evaluation, we follow the two-round dialogue procedure, and the agent stops once it predicts the `STOP` action.

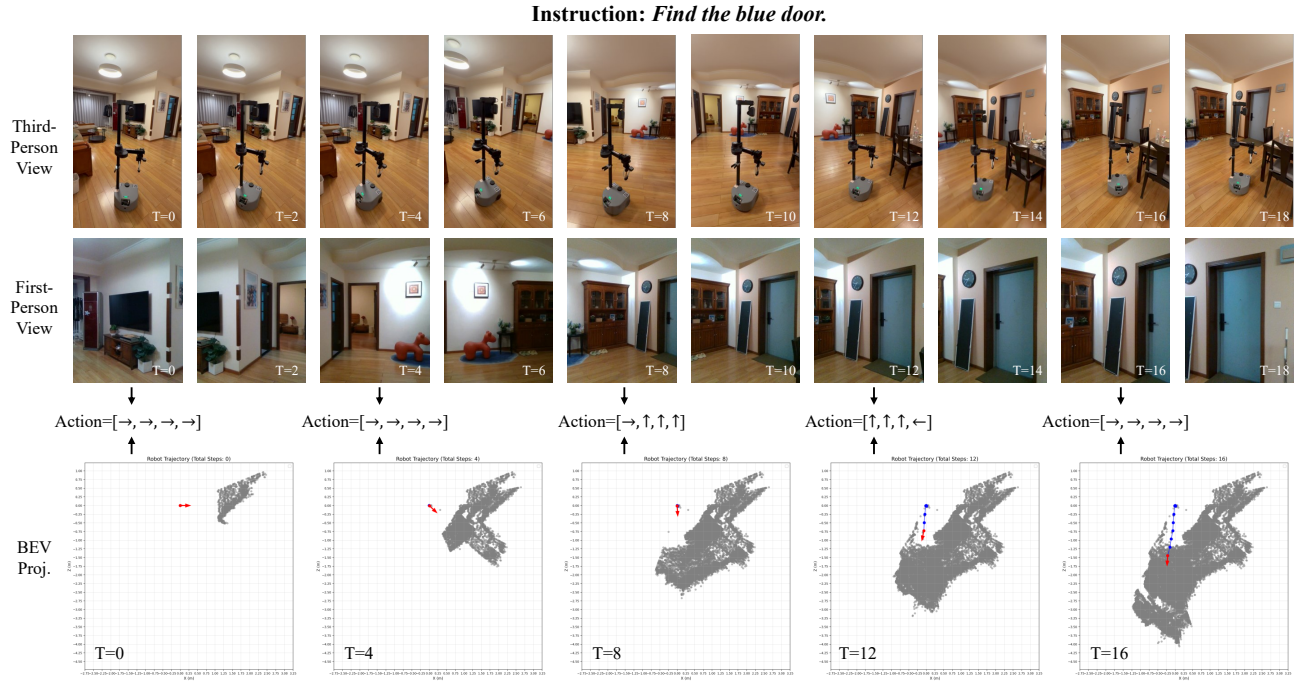


Figure 4. Real-world VLN results of GA-VLN: Example #S1.

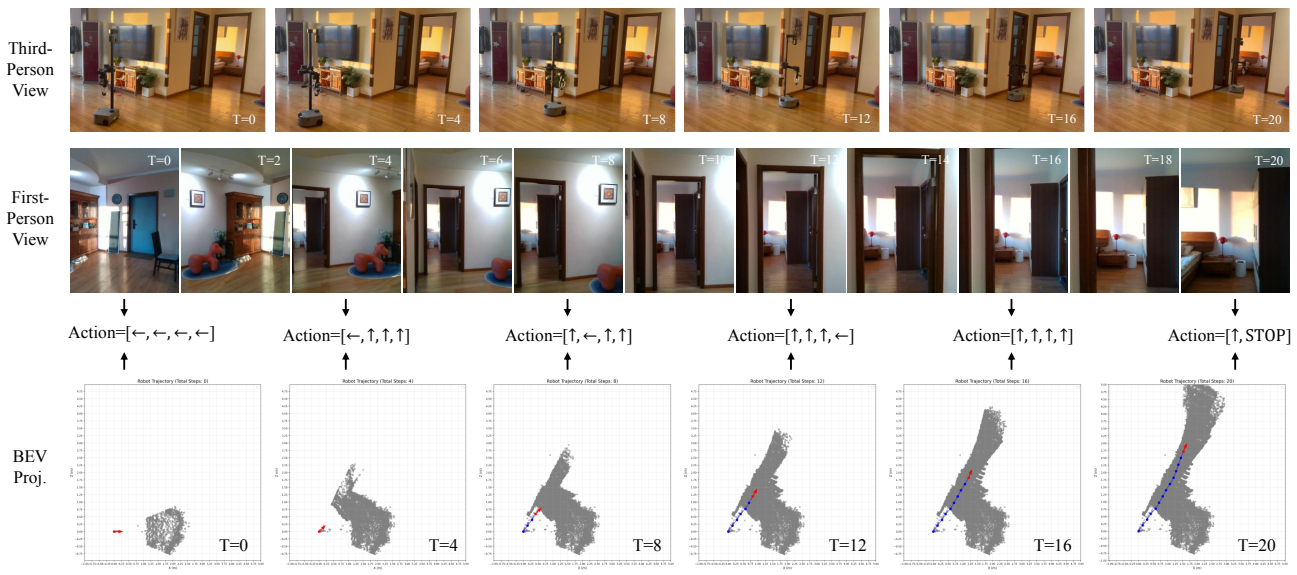


Figure 5. Real-world VLN results of GA-VLN: Example #S2.

Training Datasets	R2R-CE				RxR-CE				NavRAG-CE			
	NE↓	OSR↑	SR↑	SPL↑	NE↓	OSR↑	SR↑	SPL↑	NE↓	OSR↑	SR↑	SPL↑
w/o NavRAG-CE	4.80	**67.6**	**61.0**	**55.2**	**5.88**	**67.0**	**55.4**	45.2	**7.88**	46.4	**22.2**	**18.2**
with NavRAG-CE	**4.78**	63.2	57.9	53.9	5.98	64.7	54.3	**46.3**	8.38	**47.9**	20.1	16.2

Table 5. Ablation on training data composition across R2R-CE, RxR-CE, and NavRAG-CE benchmarks.

C. Additional Simulator Experiments

Ablation on Training Data Composition. Table 5 reveals a trade-off in data composition. Incorporating NavRAG-CE data for making in-domain performance comparable to the finetuned model but adversely affected generalization on R2R-CE and RxR-CE benchmarks. This performance drop is likely attributed to the distributional shift in instruction styles across datasets. Consequently, to ensure robust generalization, we excluded NavRAG-CE from the primary pre-training data, only utilize it for fine-tuning.

Effect of Fusion Strategy. Table 6 investigates the feature fusion strategy used in the our GA-BEV representation. Specifically, we compare two modes for fusing features from explicit depth-projected tokens and implicit 3D geometry tokens within the same BEV grid cell. Global mean pooling treats all tokens equally and computes their mean directly across sources, while hierarchical mean pooling first averages features of each modality separately and then fuses the two averaged representations. Results show that the global mean pooling achieves consistently better performance. We attribute this improvement to the higher resolution and richer patch-level representations of the 3D foundation model, whose pretrained geometric priors are better preserved under global fusion. We attribute this improvement to its ability to preserve the fine-grained local feature distribution and maintain a balanced contribution between explicit and implicit geometry cues. In contrast, hierarchical fusion tends to oversmooth each modality before integration, weakening local geometric variations critical for accurate navigation.

Effect of BEV Update Step. As described in Sec. 3.3, we adopt a structured navigation prompt for action prediction. The number of prompt-conversion rounds determines how frequently the BEV representation is updated during navigation. Table 7 analyzes the effect of this BEV update interval on navigation performance. This parameter jointly influences both the spatial accuracy of the BEV representation and the number of training samples. Shorter BEV update intervals yield more temporally precise representations but substantially increase training cost due to finer trajectory segmentation. Conversely, longer intervals reduce computation but lead to outdated spatial representations and weaker navigation performance. Table 7 shows that updating the BEV every 8 steps achieves the best balance between efficiency and accuracy, reducing total training time by nearly half with minimal performance loss compared with 4 updating steps.

Fusion Strategy	NE↓	OSR↑	SR↑	SPL↑
Global Mean Pooling	**5.03**	**59.60**	**53.56**	**49.41**
Hierarchical Mean Pooling	5.33	56.82	50.57	47.40

Table 6. Comparison of BEV feature fusion strategies.

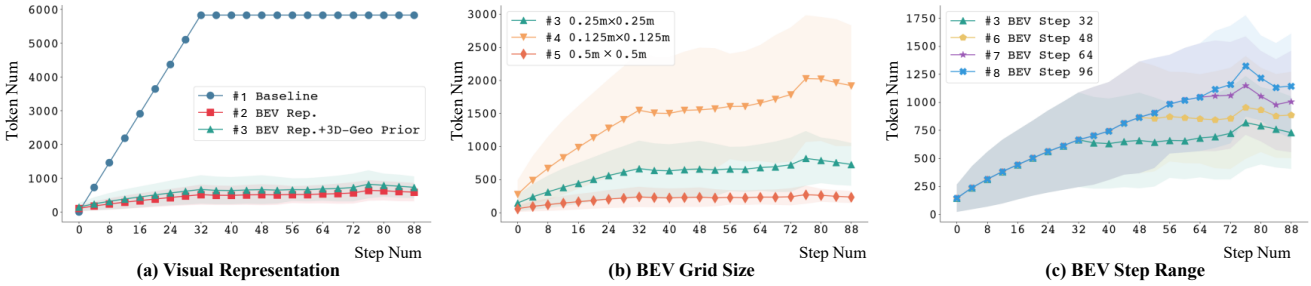
BEV Update Steps	NE↓	OSR↑	SR↑	SPL↑
4	5.37	**59.92**	**51.98**	47.18
8	**5.33**	56.33	51.50	**48.25**
12	5.79	57.48	49.37	45.15
16	6.15	54.49	46.11	41.77

Table 7. Ablation on BEV update interval w/o 3D geometry priors.

Setting	Token Num	NE↓	OSR↑	SR↑	SPL↑
RGB-Only	4003	6.08	54.59	46.49	42.36
RGB-Depth	8006	6.32	43.61	38.61	35.97
BEV Rep.	394	**5.33**	**56.33**	**51.50**	**48.25**

Table 8. Ablation on different depth processing strategies.

Ablation on Depth Processing Strategies. Unlike prior VLN methods [10, 39, 41, 42] that rely solely on RGB observations, our approach additionally incorporates depth information. Table 8 examines different strategies for processing depth and demonstrates the effectiveness and efficiency of our GA-BEV representation. The second row corresponds to a naive fusion strategy, where each RGB image is concatenated with a depth image encoded by the same visual encoder. This doubling of input tokens leads to a substantial increase in sequence length, imposing heavier computation while yielding noticeably worse performance across all metrics. In contrast, our BEV representation compresses both RGB and depth cues into a compact set of fewer tokens while achieving better performance. These results highlight that simply appending depth features is not an effective way to exploit geometric cues, whereas GA-



(a) Visual Representation **(b) BEV Grid Size** **(c) BEV Step Range**

Figure 6. Comparison of token usage across navigation steps for different configurations. The number shows in each legend corresponds to the configuration of the respective row in Table 3. The shaded area in the figure indicates the variance range of the sample data.

Nav-VQA	NE↓	OSR↑	SR↑	SPL↑
×	4.80	**67.59**	**60.96**	**55.19**
✓	**4.66**	67.37	59.92	55.05

Table 9. Ablation on the Nav-VQA Task.

BEV provides a more structured, geometry-aware representation that is both more accurate and significantly more token-efficient.

Experiments on the VQA Task. Compared with prior methods [10, 39] that rely on large-scale general-purpose VQA datasets for co-training, our model does not incorporate any additional non-navigation data in order to maintain reasonable computational cost. Nevertheless, we conduct an auxiliary Nav-VQA experiment using only the navigation datasets. Following [10, 42], the prompt is formatted as: *"User: Assume you are a robot designed for navigation. You are provided with captured image sequences* `<Images>`*. Based on this image sequence, please describe the navigation trajectory of the robot. Assistant:* `<instruction>`*"*, where `<Images>` denotes 8 sampled RGB frames along the trajectory and `<instruction>` is the navigation instruction. As shown in Table 9, incorporating Nav-VQA supervision yields only marginal changes across all evaluation metrics. This suggests that our navigation datasets are sufficiently large and of high quality, enabling the MLLM to develop robust multimodal reasoning capabilities without requiring additional VQA-style data.

Visual Token Efficiency Across Navigation Steps. To further illustrate the computational efficiency of our proposed GA-BEV representation, Figure 6 visualizes the accumulation of visual tokens across navigation steps for the various configurations detailed in Table 3. As the agent explores the environment, the standard image-based baseline (Configuration #1) exhibits a rapid, near-linear growth in token usage, quickly leading to an overwhelming computational burden and risking exceeding the MLLM's context window. In stark contrast, our GA-VLN variants (e.g., Configurations #2 and #3) maintain a remarkably low and stable token footprint throughout the entire long-horizon navigation trajectory. The shaded regions, which indicate the variance range of the sample data, further demonstrate the robustness and stability of our BEV updating mechanism across different environments and episode lengths. Additionally, the token accumulation trends for other hyperparameter settings (i.e., varying BEV grid sizes and step ranges corresponding to Rows #4 - #8) are also provided in the figure for comprehensive reference. Ultimately, this visualization confirms that compressing historical observations into a compact, ego-centric BEV space effectively eradicates the visual token redundancy issue, thereby enabling efficient and sustainable spatial reasoning for MLLMs in continuous environments.