GEOSCAFFOLD: LEARNING COMPACT GEOMETRIC LATENTS VIA RECONSTRUCTION FOR EFFICIENT VISION-LANGUAGE NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent vision-and-language navigation (VLN) systems increasingly adopt streaming Video-LLM policies that map egocentric RGB observations and instructions directly to low-level actions. Yet these policies inherit weak 3D geometric priors from 2D pretraining. Existing geometry-aware extensions charge a persistent inference-time price: depth sensors, 3D encoders, or per-step perception tool calls. We propose GeoScaffold, a geometric supervision framework that pays this price once, at training time, by internalizing geometry into the policy itself. It first learns a compact depth tokenizer on depth maps from the training trajectories and freezes it. It then fine-tunes the policy with a handful of learnable geometry query tokens, training their hidden states to reconstruct navigation-critical geometry such as depth, connectivity, and traversability. This supervision turns the query states into compact geometric latents for action decoding, and through the shared weights also internalizes geometry into the backbone's own representations. Like a scaffold, the tokenizer, target generators, and reconstruction heads are discarded after training, leaving the backbone and action interface unchanged. Extensive experiments show that GeoScaffold consistently outperforms leading vision-only navigators on continuous VLN benchmarks, offering a practical paradigm for lightweight edge deployment of spatially aware embodied navigation models.

1 INTRODUCTION

Vision-and-language navigation (VLN) requires an embodied agent to follow natural-language instructions while moving through a previously unseen environment (Anderson et al., 2018; Ku et al., 2020; Krantz et al., 2020). Recent progress has increasingly moved VLN toward an end-to-end vision-language-action (VLA) paradigm, in which a Video-LLM policy maps egocentric RGB observations and the instruction to navigation actions step by step (Zhang et al., 2024b; Cheng et al., 2024; Wei et al., 2025; Zhang et al., 2026a; Xin et al., 2026b). However, the backbones underlying these policies are pretrained almost entirely on internet-scale 2D data, which makes them strong at semantic recognition yet weak at metric and topological reasoning (Chen et al., 2024; Fu et al., 2024; Yang et al., 2025). Navigation demands exactly these abilities. Beyond recognizing objects and matching them to linguistic landmarks, an agent must reason about free space, relative scale, doorway structure, and route connectivity. These policies therefore struggle with depth-sensitive decisions, such as judging whether a landmark is nearby, whether a passage is traversable, or whether to continue exploring rather than stop prematurely.

Existing methods compensate for this weakness with additional machinery at inference time. Geometry-enhanced methods give the policy explicit spatial support, running a frozen 3D encoder alongside the backbone (Zeng et al., 2026; Fan et al., 2026; Liu et al., 2026) or lifting RGB-D observations into geometry-aware BEV maps (Yang et al., 2026). As shown in Figure 1(b), these components remain active at deployment, with depth sensors adding hardware requirements and 3D encoders increasing GPU memory usage and inference latency. Other methods restructure the system, adopting dual-system designs in which a slow reasoning module guides a fast controller (Wei et al., 2026; Intern Robotics, 2025; Xue et al., 2026) or zero-shot pipelines of LLM planners, perception, and path-planning components (Zhou et al., 2024; Long et al., 2025; Chen et al., 2025). Such designs require coordination among multiple modules, complicating system integration and deployment.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

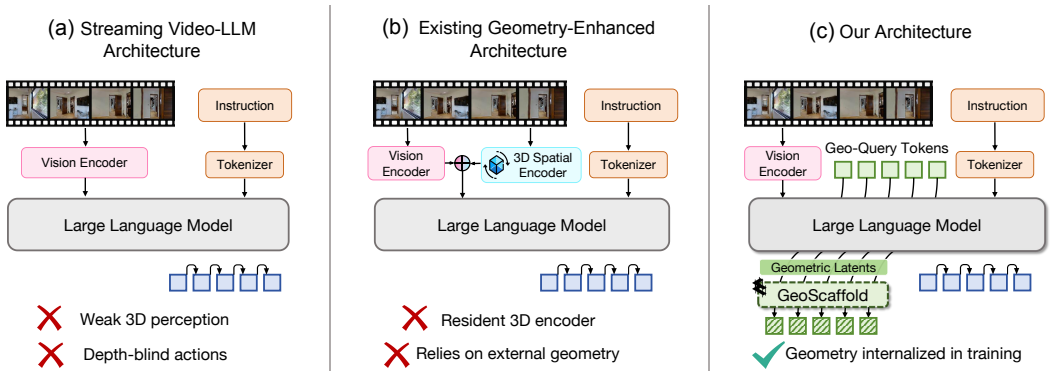

Figure 1: Comparison of end-to-end VLN paradigms. (a) Streaming Video-LLMs act directly from RGB, but lack 3D perception. (b) Geometry-enhanced policies add a spatial encoder, but it stays resident at inference and geometry is never internalized. (c) GeoScaffold internalizes geometry via training-only query-token supervision, leaving the deployed RGB-and-text action path unchanged.

By contrast, humans act with rich spatial awareness from visual experience alone, without explicit depth sensing at every step. For navigation, such spatial awareness should help the agent judge distances, identify traversable regions, and recognize passages that connect spaces. We therefore ask whether geometric supervision can equip an end-to-end VLA navigator with these abilities during training, while requiring only RGB observations and language instructions at deployment.

To this end, we propose **GeoScaffold**, a training-only geometric enhancement framework for streaming VLN. GeoScaffold supervises the policy with three geometric targets. Depth captures metric scale, while connectivity and traversability mark route-relevant passages and free space, respectively. The supervision enters through a small set of learnable geometry query tokens (Carion et al., 2020; Li et al., 2023; Mu et al., 2023; Lai et al., 2024), inserted before each action block, one dedicated group per target. Each group learns to reconstruct its own target from its hidden states. Training proceeds in two stages. Stage I learns a compact depth tokenizer offline. Stage II then fine-tunes the policy with the navigation objective unchanged plus the auxiliary reconstruction losses. This pressure compresses scene geometry into the query hidden states, which act as compact geometric latents for action decoding. After training, all auxiliary modules are discarded. The deployed policy runs on RGB and text alone, with only the learned geometry tokens remaining in the prompt.

Our contributions are threefold:

- We introduce **GeoScaffold**, a lightweight, removable geometric enhancement framework for efficient end-to-end streaming VLN that requires no depth sensors, 3D encoders, or reconstruction modules at inference time and deploys on edge hardware.
- We design a query-token reconstruction objective whose dedicated depth, connectivity, and traversability groups shape their hidden states into compact geometric latents.
- We present an empirical study of training-only geometric supervision for VLN, showing consistent success-rate gains across different backbones and in real-world deployment, with only the learned geometry tokens retained at inference.

2 Related Work

2.1 VLA Model for Navigation

Recent VLN research has increasingly reformulated navigation as an end-to-end vision-language-action problem. Navigators of this kind fine-tune a multimodal language model to map visual observations and the instruction to navigation decisions (Zhang et al., 2024b; Cheng et al., 2024; Zheng et al., 2024; Zhang et al., 2024a; 2026a), inheriting rich visual priors and the commonsense knowledge of the backbone. To run over long trajectories, other work turned the interaction into a streaming multi-turn dialogue whose cached context and compact memory keep the cost of each step bounded (Wei et al., 2025; Xin et al., 2026b). Building on these policies, dual-system designs add a

slow reasoning module above a fast controller (Wei et al., 2026; Intern Robotics, 2025; Xue et al., 2026). However, their backbones are pretrained almost exclusively on 2D web images and videos, so the resulting policies inherit little sense of metric scale or 3D structure, precisely the abilities that navigation requires.

2.2 Geometry-Enhanced VLN

A broad line of VLN research addresses this spatial weakness by injecting depth or 3D structure into the agent. These methods make geometry an input to the policy at deployment time, whether through RGB-D or stereo observations (Liu et al., 2025; Wang et al., 2026), 3D-language feature fields and tokens (Wang & Lee, 2025; Wang et al., 2025c), map-like memories (Wang et al., 2023a; Zhang et al., 2025a; 2026b; Yang et al., 2026), frozen 3D foundation models (Zeng et al., 2026; Fan et al., 2026; Liu et al., 2026) such as VGGT (Wang et al., 2025b), or agentic perception tools that query occupancy, LiDAR, or depth on demand (Xin et al., 2026a; Yue et al., 2026). These designs improve spatial understanding, but the sensors, encoders, maps, or tool calls that supply geometry must stay active at deployment and add latency, memory, or hardware to the robot. GeoScaffold instead uses geometry only as a training-time supervision signal, so the policy learns geometric representations from RGB alone and carries them into deployment as compact latents, with no geometry module on the inference path.

2.3 Auxiliary Perception Supervision and Query Tokens

Auxiliary perception supervision enriches visual representations through prediction and reconstruction tasks. Early navigation methods explore depth prediction (Mirowski et al., 2017) and self-supervised reasoning (Zhu et al., 2020). Recent approaches extend this idea to multimodal LLMs and VLAs: ROSS (Wang et al., 2025a), 3DRS (Huang et al., 2025), and Spatial Forcing (Li et al., 2026a) use visual reconstruction or 3D feature alignment, while DreamVLA (Zhang et al., 2025b) predicts future world knowledge for manipulation. Query-based aggregation and special-token readouts, as in DETR (Carion et al., 2020), BLIP-2 (Li et al., 2023), and LISA (Lai et al., 2024), provide compact interfaces for visual prediction. Perception Tokens (Bigverdi et al., 2025) and Perceptio (Li et al., 2026b) also explore autoregressive perception-token generation. GeoScaffold uses dedicated query states to reconstruct current-frame depth, connectivity, and traversability during training, learning compact geometric latents for RGB-only streaming navigation.

3 Method

3.1 Preliminaries

We study vision-and-language navigation in continuous environments (VLN-CE) (Krantz et al., 2020), in which an agent follows a natural-language instruction using egocentric RGB observations and predicts actions from the action space $\mathcal{A} = \{\textsc{Forward}\ 25\,\mathrm{cm}, \textsc{Left}\ 15^\circ, \textsc{Right}\ 15^\circ, \textsc{Stop}\}$. Streaming end-to-end VLN is commonly formulated as a multi-turn dialogue with a Video-LLM: each user turn provides the current observation with the instruction and the retained history, and each assistant turn emits an action before the next observation arrives. A Video-LLM navigation policy comprises a vision encoder $\mathcal{V}$ that tokenizes the retained RGB observations, a projector ϕ that maps visual tokens into the LLM embedding space, and an autoregressive language model $\mathcal{F}_{\mathrm{LLM}}$ that processes the joint vision-language-action sequence. Let $\mathcal{W}_t$ denote the visual window retained at turn t, let $I_{\mathcal{W}_t}$ denote the corresponding RGB observations, which are encoded into $N_v(t)$ visual tokens, and let P_t denote the $N_p(t)$ text tokens containing the instruction, dialogue markers, and previous actions. The hidden states for action generation are

$$H_t^{\mathrm{LLM}} = \mathcal{F}_{\mathrm{LLM}}([\phi(\mathcal{V}(I_{\mathcal{W}_t}))\,; P_t]) \in \mathbb{R}^{(N_v(t)+N_p(t))\times d}. \tag{1}$$

3.2 Model Architecture

Prior geometry-enhanced navigators consume geometry through dedicated spatial encoders and fusion modules that make it usable to the LLM (Zeng et al., 2026; Yang et al., 2026). This machinery then stays resident at inference. We therefore reverse the interface. Rather than consuming geometry as an

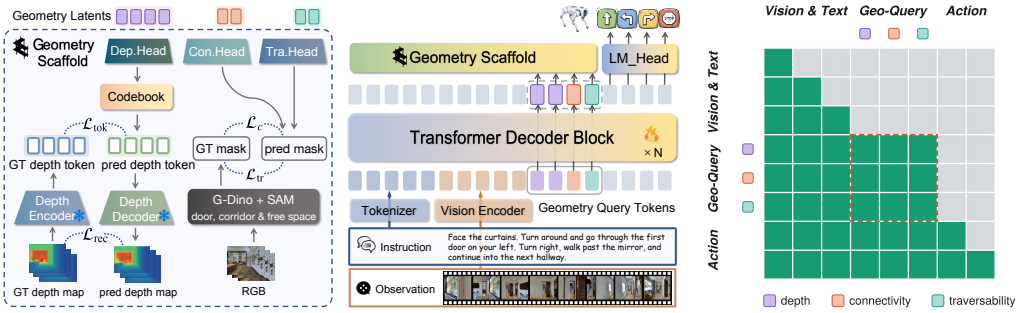

Figure 2: An overview of GeoScaffold. (a) Three geometric targets, namely depth tokens, a connectivity mask, and a traversability mask, supervise the hidden states of the query tokens through training-only reconstruction heads. (b) The hybrid attention mask, where the three query groups of one turn form a bidirectional block and all other entries stay causal.

input, the navigator learns to predict it, compressing scene geometry into a small set of geometric latents that action decoding can attend to. The latents are supervised with discrete depth tokens and binary connectivity and traversability masks. Such classification-style targets are the kind of supervision the backbone is already trained on, in contrast to dense regression signals.

Geometry query tokens. We add $K{=}K_d{+}K_c{+}K_{\mathrm{tr}}$ new vocabulary tokens $\langle \mathrm{geo}_1 \rangle, \ldots, \langle \mathrm{geo}_K \rangle$, partitioned into three groups dedicated to depth, connectivity, and traversability, respectively. These tokens are inserted at the end of every user turn, immediately before the assistant's action block, extending the input sequence of Equation (1) to

$$X_t = \big[\, \phi(\mathcal{V}(I_{\mathcal{W}_t}))\,;\; P_t;\; \langle \mathrm{geo}_1 \rangle; \cdots ; \langle \mathrm{geo}_K \rangle \,\big], \qquad H_t^{\mathrm{LLM}} = \mathcal{F}_{\mathrm{LLM}}(X_t) \in \mathbb{R}^{(N_v(t)+N_p(t)+K)\times d}. \tag{2}$$

The query tokens are excluded from the language-modeling loss and never appear in generated text, while their input and output embeddings remain trainable. Because they precede the action block, subsequent action tokens can attend to them through the causal part of the attention mask. The hidden states at the query positions realize the geometric latents introduced above, carrying a compact geometric summary of the current observation into action decoding.

Hybrid attention for query tokens. Standard causal masking would impose an arbitrary left-to-right order on the K query slots, which jointly summarize a single observation and carry no intrinsic ordering. We therefore unmask each turn's query group internally. Let $\mathbf{A} \in \{0,1\}^{L_t \times L_t}$ denote the attention mask over X_t with $L_t = N_v(t) + N_p(t) + K$, and let $\mathcal{Q}^{(s)}$ denote the positions of the query tokens inserted at turn s. All non-query positions follow the standard causal rule, and for a query position $i \in \mathcal{Q}^{(s)}$ we additionally unmask its own group:

$$\mathbf{A}_{ij} = 1 \quad \Longleftrightarrow \quad j \le i \;\text{ or }\; j \in \mathcal{Q}^{(s)}. \tag{3}$$

Query tokens of a turn attend to one another in both directions, and other positions stay causal.

Geometric supervision as a scaffold. At each supervised turn t, we build one target per type $g \in \mathcal{G} = \{d, c, \mathrm{tr}\}$, namely depth codes, a connectivity mask, and a traversability mask:

$$\tau_t^d = \mathcal{E}(D_t) \in \{1, \ldots, C\}^N, \qquad m_t^c = \mathrm{vec}(\mathcal{C}_t) \in \{0,1\}^M, \qquad m_t^{\mathrm{tr}} = \mathrm{vec}(\mathcal{T}_t) \in \{0,1\}^M, \tag{4}$$

where $M = H_g W_g$. The depth target τ_t^d is the output of the frozen tokenizer, computed without gradients. The traversability map $\mathcal{T}_t$ marks free-space and floor-like regions through which the agent can move, while the connectivity map $\mathcal{C}_t$ marks passages relevant to route continuation, such as doors, corridors, and openings. Both maps are generated offline from the egocentric RGB frames with Grounding-DINO (Liu et al., 2024) and SAM (Kirillov et al., 2023), as detailed in Appendix C.2.

Let $\mathcal{Q}_g$ denote the set of query-token positions assigned to geometry type g, and let $K_g = |\mathcal{Q}_g|$. We gather the corresponding last-layer hidden states from H_t^{LLM}:

$$Z_t^g = H_t^{\mathrm{LLM}}[\mathcal{Q}_g] \in \mathbb{R}^{K_g \times d}, \qquad g \in \mathcal{G}. \tag{5}$$

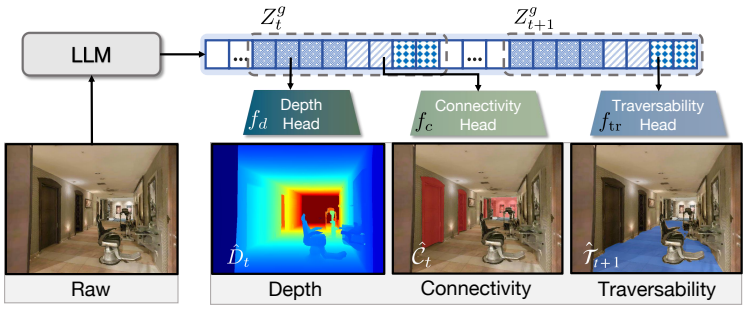

Figure 3: Depth, connectivity, and traversability targets decoded from the query-token interface. Traversability highlights free or floor-like regions, while connectivity emphasizes passages and openings relevant to route continuation.

Each group of query states is flattened and decoded by a target-specific reconstruction head:

$$\hat{\tau}_t^d = f_d(\mathrm{vec}(Z_t^d)) \in \mathbb{R}^{N\times C}, \quad \hat{m}_t^c = f_c(\mathrm{vec}(Z_t^c)) \in \mathbb{R}^M, \quad \hat{m}_t^{\mathrm{tr}} = f_{\mathrm{tr}}(\mathrm{vec}(Z_t^{\mathrm{tr}})) \in \mathbb{R}^M. \quad (6)$$

These are logits: $\hat{\tau}_t^d$ holds one logit per codebook entry at each depth-token position, while $\hat{m}_t^c$ and $\hat{m}_t^{\mathrm{tr}}$ hold one logit per pixel of the connectivity and traversability masks. We write $\hat{\mathcal{C}}_t$ and $\hat{\mathcal{T}}_t$ for these predictions reshaped back to $H_g \times W_g$. Thus, four query tokens jointly predict the N discrete VQ depth codes in a single forward pass, two query tokens predict the binary connectivity mask, and two query tokens predict the binary traversability mask. The depth codes are not added to the language vocabulary and are never generated autoregressively.

Removing the scaffold at inference. After training, the reconstruction heads and the depth tokenizer are removed. The deployed policy is the original backbone with the learned query-token embeddings in the prompt and the hybrid mask of Equation (3), a masking pattern that adds no parameters or computation. At inference the navigator receives only RGB, the instruction, and the dialogue history, and the acquired geometry reaches action decoding through the geometric latents at the query positions.

3.3 Two-Stage Training Strategy

Stage I: depth tokenizer training. The first stage learns a discrete, decodable representation of per-frame geometry. We train a small depth VQ-VAE (van den Oord et al., 2017) offline on normalized depth maps from the navigation *training* trajectories. The tokenizer is never exposed to evaluation environments. The convolutional encoder $\mathcal{E}$ maps a depth map D_t to a grid of N latents, each of which is quantized against a C-entry codebook $\mathbf{B}$, yielding the depth code indices τ_t^d. The convolutional decoder $\mathcal{D}$ reconstructs the depth map from the corresponding code embeddings. The tokenizer is trained with the standard VQ-VAE objective, comprising reconstruction, codebook, and commitment terms. After this stage, $\mathcal{E}$, $\mathcal{D}$, and $\mathbf{B}$ are frozen. During policy training, $\mathcal{E}$ serves only as a target generator, and $\mathcal{D}$ serves only as a fixed differentiable renderer for the soft reconstruction loss.

Stage II: geometry-scaffolded policy training. The second stage fine-tunes the navigation policy with the query tokens and hybrid mask of Section 3.2. Alongside the unchanged imitation objective $\mathcal{L}_{\mathrm{policy}}$, the reconstruction heads decode the query hidden states and are supervised against the frozen targets of Equation (4) through three losses. The depth-token loss $\mathcal{L}_{\mathrm{tok}}$ is the mean cross-entropy over the N depth-token positions. To supervise decodable geometry beyond token identity, the reconstruction loss $\mathcal{L}_{\mathrm{rec}}$ renders the predicted code distribution through the frozen decoder and compares the result with the ground-truth depth map. The two losses are

$$\mathcal{L}_{\mathrm{tok}} = -\frac{1}{N}\sum_{n=1}^{N} \log \mathrm{softmax}\left(\hat{\tau}_t^{d,n}\right)_{\tau_t^{d,n}}, \qquad \mathcal{L}_{\mathrm{rec}} = \left\|\mathcal{D}\left(\mathrm{softmax}(\hat{\tau}_t^d)\,\mathbf{B}\right) - D_t\right\|_2^2, \quad (7)$$

where $\hat{\tau}_t^{d,n} \in \mathbb{R}^C$ denotes the predicted code logits and $\tau_t^{d,n}$ the code index produced by the frozen tokenizer at depth-token position n. Gradients propagate through the frozen decoder to the

reconstruction head and the Video-LLM, while the decoder parameters remain fixed. For each mask target $g \in \{c, \mathrm{tr}\}$, with target m_t^g and predicted logits $\hat{m}_t^g$, we employ a binary mask reconstruction loss that combines pointwise classification with a soft region-overlap term:

$$\mathcal{L}_g = \mathrm{BCE}\big(\hat{m}_t^g,\, m_t^g\big) + \lambda_{\mathrm{dice}}\, \mathcal{L}_{\mathrm{Dice}}\big(\sigma(\hat{m}_t^g),\, m_t^g\big), \tag{8}$$

where $\sigma(\cdot)$ is the sigmoid. The objective adds the geometric reconstruction losses to the imitation loss:

$$\mathcal{L} = \mathcal{L}_{\mathrm{policy}} + \lambda_{\mathrm{tok}}\, \mathcal{L}_{\mathrm{tok}} + \lambda_{\mathrm{rec}}\, \mathcal{L}_{\mathrm{rec}} + \lambda_c\, \mathcal{L}_c + \lambda_{\mathrm{tr}}\, \mathcal{L}_{\mathrm{tr}}. \tag{9}$$

4 Experiments

4.1 Experimental Setup

Datasets & evaluation metrics. We train and evaluate in the VLN-CE setting (Krantz et al., 2020), where trajectories are executed as low-level actions in Matterport3D scenes (Chang et al., 2017). Our navigation-action data consists of oracle trajectories from R2R-CE and RxR-CE, extended with the extra navigation data and DAgger trajectories of the StreamVLN recipe (Wei et al., 2025). Every trajectory frame is additionally paired with generated depth, connectivity, and traversability annotations (Section 3.2), which serve only as training-time reconstruction targets. We report the standard VLN-CE metrics, namely Navigation Error (NE), Success Rate (SR), Oracle Success Rate (OS), Success weighted by Path Length (SPL), and normalized Dynamic Time Warping (nDTW), among which SR and SPL are the widely adopted primary measures of task completion and efficiency.

Implementation details. We adopt the streaming VLN formulation of StreamVLN (Wei et al., 2025), a multi-turn dialogue with slow-fast visual context, and instantiate it on Qwen3-VL-4B (Bai et al., 2025), decoding one action per turn. The geometry branch uses $K{=}8$ query tokens ($K_d{=}4$, $K_c{=}K_{\mathrm{tr}}{=}2$) read from the last hidden layer, and the depth tokenizer is a 0.74M-parameter VQ-VAE with $N{=}100$ tokens over a $C{=}128$-entry codebook. Throughout, a policy without GeoScaffold denotes the same backbone trained with the identical recipe and data but without the query tokens, reconstruction heads, and geometric losses. Training takes approximately 1500 A100 GPU hours for the policy and around 50 for the VQ-VAE. Further implementation details are given in Appendix C.1.

4.2 Main Results

Results on VLN-CE benchmarks. Table 1 compares GeoScaffold with published systems on R2R-CE and RxR-CE Val-Unseen. Rows marked with * use only R2R-CE and RxR-CE for navigation training, whereas the final block draws on large-scale additional navigation data such as DAgger trajectories and augmented corpora. Two observations stand out. First, geometric supervision is data-efficient. Trained on VLN-CE data alone, GeoScaffold-4B* reaches 55.3% SR and 51.0% SPL on R2R-CE, ahead of JanusVLN-7B* under the same data, and in SR and SPL it already exceeds NaVILA-7B on both benchmarks and matches or exceeds StreamVLN-7B and InternVLA-N1 on RxR-CE, although these use more navigation data. With the additional data, GeoScaffold-4B reaches 64.8% SR and 60.5% SPL on R2R-CE, exceeding JanusVLN-7B by 4.3 and 3.7 points in SR and SPL and the RGB-D policy GA-VLN-7B by 3.8 and 5.3 points while observing only RGB. Second, GeoScaffold-4B achieves higher SR and SPL on both benchmarks than the 7B VLA systems listed in Table 1. On RxR-CE, whose instructions are longer and more fine-grained, it reaches 59.6% SR, 52.2% SPL, and 66.3 nDTW and improves over NavFoM-7B by 2.2 and 2.8 points in SR and SPL.

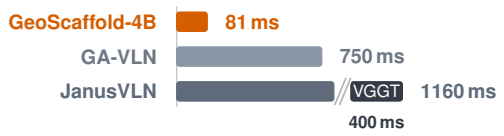

Figure 4: Average per-frame latency.

Inference efficiency against geometry-enhanced methods. We benchmark inference on a single NVIDIA RTX 4090 and report the average wall-clock latency to process one incoming frame and emit its action. All measurements are taken in the steady state, after the streaming context has filled the sliding window and the per-frame cost has stabilized. As shown in Figure 4, GeoScaffold-4B processes a frame in 81 ms. GA-VLN (Yang et al., 2026), which lifts RGB-D observations into a geometry-aware BEV map enriched with 3D foundation-model features, requires 750 ms, and JanusVLN (Zeng et al., 2026), which runs a resident VGGT 3D encoder at every step, requires

Table 1: Comparison with published systems on R2R-CE and RxR-CE Val-Unseen. Variants marked with * use only R2R-CE and RxR-CE for navigation training. StreamVLN† uses EnvDrop-augmented navigation data, while NaVILA† excludes human touring videos.

Method	Observation				R2R-CE Val-Unseen				RxR-CE Val-Unseen			
	Pano.	Odo.	Depth	S.RGB	NE↓	OS↑	SR↑	SPL↑	NE↓	SR↑	SPL↑	nDTW↑
CMA (Hong et al., 2022)	✓	✓	✓		6.20	52.0	41.0	36.0	8.76	26.5	22.1	47.0
Sim2Sim (Krantz & Lee, 2022)	✓	✓	✓		6.07	52.0	43.0	36.0	–	–	–	–
VLN↻BERT (Hong et al., 2022)	✓	✓	✓		5.74	53.0	44.0	39.0	8.98	27.0	22.6	46.7
GridMM (Wang et al., 2023a)	✓	✓	✓		5.11	61.0	49.0	41.0	–	–	–	–
Reborn (An et al., 2022)	✓	✓	✓		5.40	57.0	50.0	46.0	5.98	48.6	42.0	63.3
LAW (Raychaudhuri et al., 2021)		✓	✓	✓	6.83	44.0	35.0	31.0	10.90	8.0	8.0	38.0
Seq2Seq (Krantz et al., 2020)			✓	✓	7.77	37.0	25.0	22.0	12.10	13.9	11.9	30.8
NaVid-4D (Liu et al., 2025)			✓	✓	5.99	55.7	43.8	37.1	–	–	–	–
NavMorph (Yao et al., 2025)			✓	✓	5.75	56.9	47.9	33.2	–	–	–	–
GA-VLN-7B (Yang et al., 2026)		✓	✓	✓	4.80	67.6	61.0	55.2	5.88	55.4	45.2	–
NaVid-7B (Zhang et al., 2024b)				✓	5.47	49.1	37.4	35.9	–	–	–	–
StreamVLN-7B† (Wei et al., 2025)				✓	6.05	53.8	45.5	41.6	–	–	–	–
Uni-NaVid-7B (Zhang et al., 2024a)				✓	5.58	53.3	47.0	42.7	6.24	48.7	40.9	–
NaVILA-7B† (Cheng et al., 2024)				✓	5.37	57.6	49.7	45.5	–	–	–	–
JanusVLN-7B* (Zeng et al., 2026)				✓	5.17	58.0	52.8	49.2	6.46	51.4	44.3	59.1
GeoScaffold-4B* (Ours)				✓	**5.07**	**61.9**	**55.3**	**51.0**	**5.82**	**53.2**	**46.2**	**59.8**
NaVILA-7B (Cheng et al., 2024)				✓	5.22	62.5	54.0	49.0	6.77	49.3	44.0	58.8
StreamVLN-7B (Wei et al., 2025)				✓	4.98	64.2	56.9	51.9	6.22	52.9	46.0	61.9
InternVLA-N1 (Intern Robotics, 2025)				✓	4.89	60.6	55.4	52.1	6.41	49.5	41.8	62.6
NavFoM-7B (Zhang et al., 2026a)				✓	5.01	64.9	56.2	51.2	5.51	57.4	49.4	60.2
JanusVLN-7B (Zeng et al., 2026)				✓	4.78	65.2	60.5	56.8	6.06	56.2	47.5	62.1
GeoScaffold-4B (Ours)				✓	**4.17**	**72.1**	**64.8**	**60.5**	**4.51**	**59.6**	**52.2**	**66.3**

1,160 ms. The two geometry-enhanced policies are therefore $9\times$ to $14\times$ slower than GeoScaffold-4B. The VGGT encoder alone accounts for 400 ms, almost $5\times$ the per-frame inference of GeoScaffold-4B. Training-time geometry thus buys spatial competence without the standing inference cost that resident geometry modules impose, and GeoScaffold-4B sustains control rates above 12 Hz on a single consumer GPU. Per-frame latency traces are given in Appendix B.

Backbone generality. Because GeoScaffold changes only the supervision and modifies no architecture, sensor, or action interface, porting it to a new backbone amounts to re-running training. We apply the identical recipe to three backbones from three model families, namely LLaVA-Video-7B, Qwen2.5-VL-3B, and Qwen3-VL-4B, and train each of them without and with GeoScaffold on R2R-CE and RxR-CE only, the data configuration used for all ablations. Table 2 reports the paired results. GeoScaffold improves every backbone on all four metrics. SR rises by 5.5 to 7.9 points, SPL by 6.4 to 8.7 points, and NE falls by 0.5 to 0.8 m. The gains hold across three model families and model sizes from 3B to 7B without any change to the recipe, so GeoScaffold transfers to a new backbone as a drop-in training procedure.

Table 2: Backbone generality. Each backbone is trained with the identical recipe without and with GeoScaffold on R2R-CE and RxR-CE only, and evaluated on R2R-CE Val-Unseen.

Backbone	w/o GeoScaffold				w/ GeoScaffold				ΔSR
	NE↓	OS↑	SR↑	SPL↑	NE↓	OS↑	SR↑	SPL↑	
LLaVA-Video-7B	6.15	50.2	39.4	37.1	5.61	56.8	47.3	45.8	+7.9
Qwen2.5-VL-3B	5.96	53.7	46.4	41.6	5.14	58.5	51.9	49.2	+5.5
Qwen3-VL-4B	5.77	56.2	48.8	44.6	5.07	61.9	55.3	51.0	+6.5

4.3 Qualitative Results

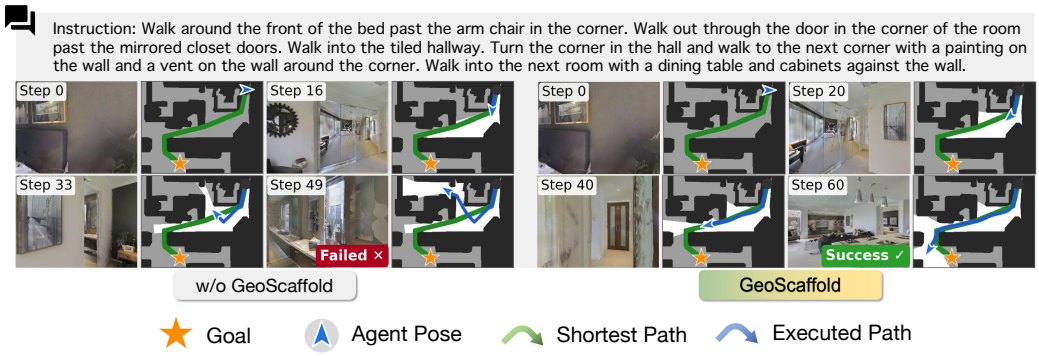

Figure 5: Simulated qualitative comparison on VLN-CE.

Simulation. Figure 5 compares closed-loop rollouts with and without GeoScaffold on an R2R-CE Val-Unseen episode from the same start pose and instruction. Without GeoScaffold the policy leaves the hallway too early and stops prematurely in an adjacent room. With GeoScaffold the policy follows the hallway around the corner and stops only in the described dining room. The rollouts thus show a policy that is more sensitive to spatial structure, to which openings continue the route and to how far it still has to go. More episodes are shown in Appendix D.2.

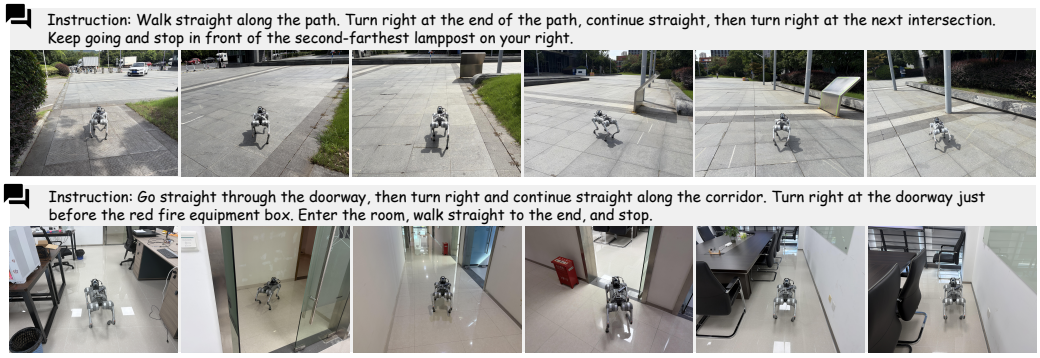

Figure 6: Zero-shot real-world qualitative results.

Real-world deployment. Policies with a resident depth or 3D encoder are costly to run on a robot and are often moved to off-board servers, where remote communication and asynchronous inference add latency and lower the control frequency. GeoScaffold carries no such cost, and the Qwen3-VL-4B policy can run locally on an NVIDIA Jetson edge device from RGB and language alone. We deploy it on a Unitree Go2 Edu quadruped with an Intel RealSense D435i camera, using only its RGB stream, and Figure 6 shows two zero-shot episodes. The geometry-demanding instructions of our real-world experiments stress three forms of spatial understanding: depth-aware reasoning toward distant landmarks such as a street lamp, traversability reasoning when objects partially block the route, and connectivity reasoning in doorway and porch-like scenes. GeoScaffold completes both episodes and raises the success rate on geometry-demanding instructions over a larger instruction set (Figure 7 in Appendix B), indicating that the spatial reasoning instilled by training-time supervision transfers to the real world. Appendix D.3 shows more episodes.

4.4 Ablation Study

Unless otherwise stated, all ablations in this section are trained without external data and evaluated on the R2R-CE Val-Unseen benchmark. Additional ablations on the query-token budget and a comparison with consuming depth as input are given in Appendix A.

Supervision-target composition. Table 3 ablates the three supervision targets reconstructed by the full model (Section 3.2) by removing one at a time. Depth is the dominant signal. Removing depth-token supervision lowers SR from 55.3 to 51.1, the largest single drop, with consistent drops

in NE, OS, and SPL. The two masks contribute smaller but clear gains on top of depth: removing traversability costs 2.2 SR and removing connectivity 2.6 SR. Every variant degrades on all four metrics, so the targets are complementary rather than redundant. The masks add passage and free-space cues that depth alone does not capture. We therefore adopt all three targets throughout.

Table 3: Ablation on supervision-target composition.

Variant	NE↓	OS↑	SR↑	SPL↑
GeoScaffold	5.07	61.9	55.3	51.0
w/o depth	5.25	57.4	51.1	47.1
w/o traversability	5.23	58.7	53.1	49.2
w/o connectivity	5.18	58.5	52.7	48.2

Table 4: Ablation on depth representation and losses.

Depth target / loss	NE↓	OS↑	SR↑	SPL↑
VQ-VAE, $\mathcal{L}_{\text{tok}}+\mathcal{L}_{\text{rec}}$ (Ours)	5.11	59.4	52.3	48.8
w/o $\mathcal{L}_{\text{tok}}$ ($\mathcal{L}_{\text{rec}}$ only)	5.93	55.2	48.5	43.7
w/o $\mathcal{L}_{\text{rec}}$ ($\mathcal{L}_{\text{tok}}$ only)	5.19	57.8	51.0	46.4
Continuous VAE latent	5.35	56.6	50.9	45.9
Pixel-space regression	5.66	55.8	49.3	44.6

Depth representation and reconstruction loss. Table 4 varies how the depth target is represented and supervised, with depth as the only supervision target. Discrete token classification carries most of the signal. Dropping $\mathcal{L}_{\text{tok}}$ costs 3.8 SR, whereas dropping $\mathcal{L}_{\text{rec}}$ costs 1.3, and the two losses together remain best. Discreteness also matters. Replacing the VQ codes with a capacity-matched continuous latent drops 1.4 SR, and pixel-space depth regression costs 3.0 SR. Compact, discrete, decodable targets are thus the right target for supervising a token-based policy.

Hidden-state source for reconstruction. Table 5 varies which positions the reconstruction heads read from, with targets and losses unchanged. Supervising image hidden states leads to the largest SR drop (12.1 points), possibly because forcing visual features to support geometric reconstruction degrades their quality for navigation. Supervising action hidden states costs 3.5 SR, as the reconstruction gradient competes with the imitation objective at the positions that generate actions. Query tokens without reconstruction fall to 47.7 SR, so the gains come from the geometric objective rather than from the extra tokens. In short, query tokens work best because they absorb the geometric supervision at positions that the backbone does not otherwise use for perception or action.

Table 5: Ablation on the hidden-state source.

Hidden-state source	NE↓	OS↑	SR↑	SPL↑
Image hidden states	6.78	50.6	43.2	38.5
Action hidden states	5.23	58.4	51.8	47.9
Query tokens (Ours)	5.07	61.9	55.3	51.0
w/o reconstruction	6.06	54.8	47.7	43.3

Table 6: Ablation on the query-token attention mask.

Attention mask	NE↓	OS↑	SR↑	SPL↑
Hybrid mask (Ours)	5.07	61.9	55.3	51.0
latents blocked at test	5.17	58.9	52.8	47.7
Causal mask	5.15	59.6	53.9	48.1
Causal, reversed order	5.20	59.8	53.5	48.7

Attention design and latent utilization. Table 6 isolates the attention design. Replacing the hybrid mask with a causal mask over the query tokens costs 1.4 SR. Reversing the query order in the causal variant changes SR only slightly, from 53.9 to 53.5, suggesting limited sensitivity to query order. The hybrid mask achieves the best performance. Blocking attention to the latents at test time, without retraining, lowers SR from 55.3 to 52.8 and SPL from 51.0 to 47.7 with the weights untouched, so action decoding does read the geometric latents at inference. The blocked variant still outperforms the policy without GeoScaffold (48.8 SR, Table 2), indicating that geometry is also internalized into the model's own representations. Appendix D.1 further analyzes where action decoding attends.

5 Conclusion

We presented GeoScaffold, a training-time supervision framework that internalizes geometry into streaming VLN policy representations. By reconstructing depth, connectivity, and traversability from dedicated query states, it learns compact geometric latents for action decoding. The auxiliary modules are discarded after training, leaving a policy that navigates from RGB and language without a resident geometry encoder. Experiments demonstrate navigation gains on R2R-CE and RxR-CE, across multiple backbones, and in real-world deployment. The 4B policy averages 81 ms per frame on an RTX 4090 and also runs on board a Jetson device. These findings support geometric reconstruction as a practical training signal for efficient, spatially aware navigation.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on research methodology and on experiment and ablation design, to assist in implementing methods (training and evaluation code, backbone porting, and dataloaders), to write data-processing scripts that clean and reformat data, and to assist in interpreting results through log analysis and summary tables. All final design decisions were made by the authors. We have not used generative AI tools to help develop theoretical models or conceptual framework, to generate synthetic data sets (all experiments use standard VLN-CE benchmarks) or for translation. The remaining required-disclosure tasks, namely formulating mathematical claims, providing critical ingredients for proofs, writing proofs, and qualitative or thematic analysis of interview or survey data, are not applicable to this work, which contains no formal mathematical claims and involves no such data. Additionally, we used generative AI tools to suggest experimental parameters, summarize and analyze existing literature, search for information, edit the paper to improve readability, and format references. We have reviewed all AI-assisted work. AI-assisted code was reviewed by the authors, gated by staged unit tests, and validated by reproducing baselines and through our full training and evaluation runs. Every reported number was traced to raw experiment logs, and every citation was checked against its bibliographic record. All AI-drafted text was reviewed and edited by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Dong An, Zun Wang, Yangguang Li, Yi Wang, Yicong Hong, Yan Huang, Liang Wang, and Jing Shao. 1st place solutions for RxR-Habitat vision-and-language navigation competition (CVPR 2022). *arXiv preprint arXiv:2206.11610*, 2022.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, 2018.

Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. ScanQA: 3D question answering for spatial scene understanding. In *CVPR*, 2022.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Mahtab Bigverdi, Zelun Luo, Cheng-Yu Hsieh, Ethan Shen, Dongping Chen, Linda Shapiro, and Ranjay Krishna. Perception tokens enhance visual reasoning in multimodal language models. In *CVPR*, 2025.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3D: Learning from RGB-D data in indoor environments. In *3DV*, 2017.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brian Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. SpatialVLM: Endowing vision-language models with spatial reasoning capabilities. In *CVPR*, 2024.

Jiaqi Chen, Bingqian Lin, Xinmin Liu, Lin Ma, Xiaodan Liang, and Kwan-Yee K. Wong. Affordances-oriented planning using foundation models for continuous vision-language navigation. In *AAAI*, 2025.

An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Bıyık, Hongxu Yin, Sifei Liu, and Xiaolong Wang. NaVILA: Legged robot vision-language-action model for navigation. *arXiv preprint arXiv:2412.04453*, 2024.

Zehua Fan, Wenqi Lyu, Wenxuan Song, Linge Zhao, Yifei Yang, Xi Wang, Junjie He, Lida Huang, Haiyan Liu, Bingchuan Sun, Guangjun Bao, Xuanyao Mao, Liang Xu, Yan Wang, and Feng Gao. PROSPECT: Unified streaming vision-language navigation via semantic–spatial fusion and latent predictive representation. *arXiv preprint arXiv:2603.03739*, 2026.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. BLINK: Multimodal large language models can see but not perceive. In *ECCV*, 2024.

Yicong Hong, Zun Wang, Qi Wu, and Stephen Gould. Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation. In *CVPR*, 2022.

Xiaohu Huang, Jingjing Wu, Qunyi Xie, and Kai Han. 3DRS: MLLMs need 3D-aware representation supervision for scene understanding. In *NeurIPS*, 2025.

Intern Robotics. InternVLA-N1: An open dual-system vision-language navigation foundation model with learned latent plans. Technical report, Shanghai Artificial Intelligence Laboratory, 2025. URL https://internrobotics.github.io/internvla-n1.github.io/static/pdfs/InternVLA_N1.pdf.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *ICCV*, pp. 4015–4026, 2023.

Jacob Krantz and Stefan Lee. Sim-2-Sim transfer for vision-and-language navigation in continuous environments. In *ECCV*, 2022.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *ECCV*, 2020.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *EMNLP*, 2020.

Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. LISA: reasoning segmentation via large language model. In *CVPR*, 2024.

Fuhao Li, Wenxuan Song, Han Zhao, Jingbo Wang, Pengxiang Ding, Donglin Wang, Long Zeng, and Haoang Li. Spatial forcing: Implicit spatial representation alignment for vision-language-action model. In *ICLR*, 2026a.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023.

Yuchen Li, Amanmeet Garg, Shalini Chaudhuri, Rui Zhao, and Garin Kessler. Perceptio: Perception enhanced vision language models via spatial token generation. *arXiv preprint arXiv:2603.18795*, 2026b.

Haoran Liu, Weikang Wan, Xiqian Yu, Minghan Li, Jiazhao Zhang, Bo Zhao, Zhibo Chen, Zhongyuan Wang, Zhizheng Zhang, and He Wang. NaVid-4D: Unleashing spatial intelligence in egocentric RGB-D videos for vision-and-language navigation. In *ICRA*, 2025. URL https://lhrrhl0419.github.io/NaVid4D/.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: marrying DINO with grounded pre-training for open-set object detection. In *ECCV*, 2024.

Xiangchen Liu, Hanghan Zheng, Jeil Jeong, Minsung Yoon, Lin Zhao, Zhide Zhong, Haoang Li, and Sung-Eui Yoon. DyGeoVLN: Infusing dynamic geometry foundation model into vision-language navigation. *arXiv preprint arXiv:2603.21269*, 2026.

Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. InstructNav: Zero-shot system for generic instruction navigation in unexplored environment. In *CoRL*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2049–2060. PMLR, 2025. URL https://proceedings.mlr.press/v270/long25b.html.

Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andrew J. Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, Dharshan Kumaran, and Raia Hadsell. Learning to navigate in complex environments. In *ICLR*, 2017.

Jesse Mu, Xiang Li, and Noah Goodman. Learning to compress prompts with gist tokens. In *NeurIPS*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/3d77c6dcc7f143aa2154e7f4d5e22d68-Abstract-Conference.html.

Sonia Raychaudhuri, Saim Wani, Shivansh Patel, Unnat Jain, and Angel X. Chang. Language-aligned waypoint (LAW) supervision for vision-and-language navigation in continuous environments. In *EMNLP*, 2021.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, 2011.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NeurIPS*, 2017.

Haochen Wang, Anlin Zheng, Yucheng Zhao, Tiancai Wang, Zheng Ge, Xiangyu Zhang, and Zhaoxiang Zhang. Reconstructive visual instruction tuning. In *ICLR*, 2025a.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotný. VGGT: visual geometry grounded transformer. In *CVPR*, 2025b.

Yunheng Wang, Yuetong Fang, Taowen Wang, Lusong Li, Kun Liu, Junzhe Xu, Zizhao Yuan, Yixiao Feng, Jiaxi Zhang, Wei Lu, Zecui Zeng, and Renjing Xu. What limits vision-and-language navigation ? *arXiv preprint arXiv:2605.13328*, 2026.

Zihan Wang and Gim Hee Lee. g3D-LF: Generalizable 3D-language feature fields for embodied tasks. In *CVPR*, 2025.

Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. GridMM: Grid memory map for vision-and-language navigation. In *ICCV*, 2023a.

Zihan Wang, Seungjun Lee, and Gim Hee Lee. Dynam3D: Dynamic layered 3D tokens empower VLM for vision-and-language navigation. In *NeurIPS*, 2025c.

Zun Wang, Jialu Li, Yicong Hong, Yi Wang, Qi Wu, Mohit Bansal, Stephen Gould, Hao Tan, and Yu Qiao. Scaling data generation in vision-and-language navigation. In *ICCV*, 2023b.

Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, Xihui Liu, and Jiangmiao Pang. StreamVLN: Streaming vision-and-language navigation via SlowFast context modeling. *arXiv preprint arXiv:2507.05240*, 2025.

Meng Wei, Chenyang Wan, Jiaqi Peng, Xiqian Yu, Yuqiang Yang, Delin Feng, Wenzhe Cai, Chenming Zhu, Tai Wang, Jiangmiao Pang, and Xihui Liu. Ground slow, move fast: A dual-system foundation model for generalizable vision-language navigation. In *ICLR*, 2026.

Zihao Xin, Wentong Li, Yixuan Jiang, Ziyuan Huang, Bin Wang, Piji Li, Jianke Zhu, Jie Qin, and Sheng-Jun Huang. AgentVLN: Towards agentic vision-and-language navigation. In *ECCV*, 2026a.

Zihao Xin, Wentong Li, Yixuan Jiang, Bin Wang, Runmin Cong, Jie Qin, and Shengjun Huang. DecoVLN: Decoupling observation, reasoning, and correction for vision-and-language navigation. In *CVPR*, 2026b.

Xinda Xue, Junjun Hu, Minghua Luo, Shichao Xie, Jintao Chen, Zixun Xie, Kuichen Quan, Wei Guo, Zedong Chu, Mu Xu, and Zhengzhou Zhu. OmniNav: A unified framework for prospective exploration and visual-language navigation. In *ICLR*, 2026.

Jiahao Yang, Zihan Wang, Xiangyang Li, Xing Zhu, Yujun Shen, Yinghao Xu, and Shuqiang Jiang. GA-VLN: geometry-aware BEV representation for efficient vision-language navigation. In *CVPR*, 2026.

Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *CVPR*, 2025.

Xuan Yao, Junyu Gao, and Changsheng Xu. NavMorph: A self-evolving world model for vision-and-language navigation in continuous environments. In *ICCV*, 2025.

Lu Yue, Yue Fan, Shiwei Lian, Yu Zhao, Jiaxin Yu, Liang Xie, and Feitian Zhang. Spatial-VLN: Zero-shot vision-and-language navigation with explicit spatial perception and exploration. *arXiv preprint arXiv:2601.12766*, 2026.

Shuang Zeng, Dekang Qi, Xinyuan Chang, Feng Xiong, Shichao Xie, Xiaolong Wu, Shiyi Liang, Mu Xu, and Xing Wei. JanusVLN: Decoupling semantics and spatiality with dual implicit memory for vision-language navigation. In *ICLR*, 2026. URL https://openreview.net/forum?id=RnuB0Nlbd5.

Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-NaVid: A video-based vision-language-action model for unifying embodied navigation tasks. *arXiv preprint arXiv:2412.06224*, 2024a.

Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. NaVid: Video-based VLM plans the next step for vision-and-language navigation. In *RSS*, 2024b.

Jiazhao Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, Yuxin Fan, Wenjun Li, Zhibo Chen, Fei Gao, Qi Wu, Zhizheng Zhang, and He Wang. Embodied navigation foundation model. In *ICLR*, 2026a.

Jiwen Zhang, Zejun Li, Siyuan Wang, Xiangyu Shi, Zhongyu Wei, and Qi Wu. SpatialNav: Leveraging spatial scene graphs for zero-shot vision-and-language navigation. *arXiv preprint arXiv:2601.06806*, 2026b.

Lingfeng Zhang, Xiaoshuai Hao, Qinwen Xu, Qiang Zhang, Xinyao Zhang, Pengwei Wang, Jing Zhang, Zhongyuan Wang, Shanghang Zhang, and Renjing Xu. MapNav: A novel memory representation via annotated semantic maps for VLM-based vision-and-language navigation. In *ACL*, 2025a.

Wenyao Zhang, Hongsi Liu, Zekun Qi, Yunnan Wang, Xinqiang Yu, Jiazhao Zhang, Runpei Dong, Jiawei He, He Wang, Zhizheng Zhang, Li Yi, Wenjun Zeng, and Xin Jin. DreamVLA: A vision-language-action model dreamed with comprehensive world knowledge. In *NeurIPS*, 2025b.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. LLaVA-Video: Video instruction tuning with synthetic data. *TMLR*, 2025c.

Duo Zheng, Shijia Huang, Lin Zhao, Yiwu Zhong, and Liwei Wang. Towards learning a generalist model for embodied navigation. In *CVPR*, 2024.

Gengze Zhou, Yicong Hong, and Qi Wu. NavGPT: Explicit reasoning in vision-and-language navigation with large language models. In *AAAI*, 2024.

Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *CVPR*, 2020.

Wanrong Zhu, Jack Hessel, Anas Awadalla, Samir Yitzhak Gadre, Jesse Dodge, Alex Fang, Youngjae Yu, Ludwig Schmidt, William Yang Wang, and Yejin Choi. Multimodal C4: An open, billion-scale corpus of images interleaved with text. In *NeurIPS Datasets and Benchmarks Track*, 2023.

A Additional Ablations

Number of query tokens. Table 7 varies the depth budget $K_d \in \{2, 4, 6, 8\}$ with the mask groups fixed at $K_c{=}K_{\text{tr}}{=}2$, and Table 8 varies the total mask budget $K_c{+}K_{\text{tr}} \in \{2, 4, 6, 8\}$, split evenly between the two masks, with $K_d{=}4$ fixed. Four query tokens are the best setting on both axes, reaching 55.3 SR, and smaller or larger budgets are worse. A compact query group is therefore enough to carry the $N{=}100$ depth codes of a frame and the two binary masks.

Table 7: Ablation on the number of depth query tokens K_d.

K_d	NE↓	OS↑	SR↑	SPL↑
2	5.25	61.0	52.7	48.4
4 (default)	5.07	61.9	55.3	51.0
6	5.10	60.7	55.0	49.8
8	5.17	60.6	53.4	49.1

Table 8: Ablation on the number of mask query tokens $K_c{+}K_{\text{tr}}$.

$K_c{+}K_{\text{tr}}$	NE↓	OS↑	SR↑	SPL↑
2 (1+1)	5.14	61.4	53.5	48.6
4 (2+2, default)	5.07	61.9	55.3	51.0
6 (3+3)	5.17	60.4	53.7	49.9
8 (4+4)	5.21	60.8	53.2	49.1

Consuming versus predicting geometry. Table 9 tests whether a Video-LLM policy should predict geometry into compact latents or consume it as input, using the same frozen depth tokenizer for both. In the input variant, the N depth codes of every frame in the current window are embedded through a $C \times d$ table and inserted after the vision span of their frame. This variant uses no query tokens or reconstruction heads and is trained only with imitation loss under the same data and streaming setup. At test time, it receives noiseless Habitat depth as oracle input. It also adds N extra tokens per turn against K for GeoScaffold. A control replaces every test-time code by one constant code.

Consuming depth as input does not help. The oracle input variant reaches 44.3 SR, 4.5 below the policy without any geometry, and the constant-code control costs only 0.5 SR more, so the policy largely ignores the depth tokens and pays for the longer context instead. A dedicated fusion module might make the codes usable, as in JanusVLN and GA-VLN (Zeng et al., 2026; Yang et al., 2026), but it would reintroduce the resident machinery that GeoScaffold removes. Predicting geometry moves the other way. Inserting the query tokens without a reconstruction loss lowers SR from 48.8 to 47.7, whereas adding the reconstruction supervision yields 55.3 SR, 6.5 above the policy without GeoScaffold and 11.0 above the oracle input variant, with no depth at inference. Training-time supervision thus beats input-side depth while using strictly less information at test time.

Table 9: Consuming versus predicting geometry with the same frozen depth tokenizer. Checkmarks denote evaluation with Habitat depth at test time.

Variant	Depth at test	Tokens / turn	NE↓	OS↑	SR↑	SPL↑
GeoScaffold-4B (Ours)		K	5.07	61.9	55.3	51.0
Qwen3-VL-4B w/o GeoScaffold		0	5.77	56.2	48.8	44.6
Query tokens w/o reconstruction		K	6.06	54.8	47.7	43.3
Depth tokens as input (oracle depth)	✓	N	6.24	53.2	44.3	40.9
constant codes at test		N	6.22	52.5	43.8	40.3

B Deployment Results

Real-world quantitative results. Figure 7 reports success rates on the Unitree Go2 Edu for the same Qwen3-VL-4B policy without and with GeoScaffold. Indoor episodes run the policy on a remote RTX 4090 over a wireless link, while outdoor episodes have no network and run it on board a Jetson Orin NX 16 GB at 546 ms per frame. Each environment uses 20 general and 20 geometry-demanding instructions, executed three times per policy from the same start pose with the two policies alternated, for 240 episodes per policy. General instructions can be solved through landmark recognition alone, whereas geometry-demanding ones require judging the distance to a far landmark, passing a partially blocked route, or traversing doorways and porch-like openings. An episode fails on collision or an operator stop and succeeds if the robot stops within 1 m of the goal

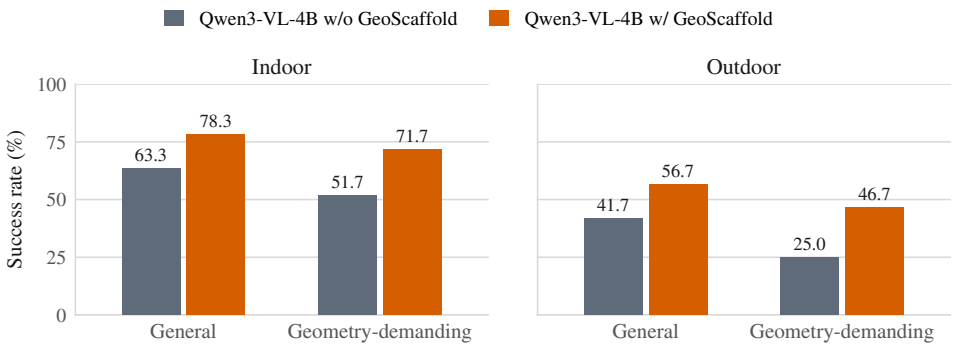

Figure 7: Real-world success rate of the same Qwen3-VL-4B policy without and with GeoScaffold on the Unitree Go2 Edu, in indoor scenes with inference on a remote RTX 4090 and in outdoor scenes with on-board inference on a Jetson Orin NX. Each bar aggregates 60 episodes, namely 20 instructions executed three times.

indoors and 3 m outdoors. GeoScaffold improves every setting, by 15.0 points on general instructions in both environments, from 63.3% to 78.3% indoors and from 41.7% to 56.7% outdoors, and by more on geometry-demanding ones, 20.0 points indoors, from 51.7% to 71.7%, and 21.7 points outdoors, from 25.0% to 46.7%. The policy without GeoScaffold loses the most on geometry-demanding instructions and GeoScaffold narrows exactly this gap, on board as well as in the cloud.

Per-frame latency traces. Figure 8 complements the averages of Figure 4 with the per-frame inference latency of all three systems over an 80-frame episode on the same RTX 4090. GeoScaffold-4B stays below 100 ms outside a short spike every eight frames, which coincides with the advance of the streaming context and the re-encoding of the history frames. JanusVLN starts near 240 ms and climbs steadily to about 1.2 s as its VGGT encoder processes a growing set of frames, and it plateaus only after roughly 56 frames. GA-VLN alternates between a lower and a higher level, following its two-round dialogue schedule in which the BEV representation is rebuilt every second query. The traces thus show that the gap in Figure 4 is not an artifact of averaging. The geometry-enhanced policies are slower at every frame, and their cost either grows with the episode or fluctuates with a rebuild schedule, whereas GeoScaffold stays flat.

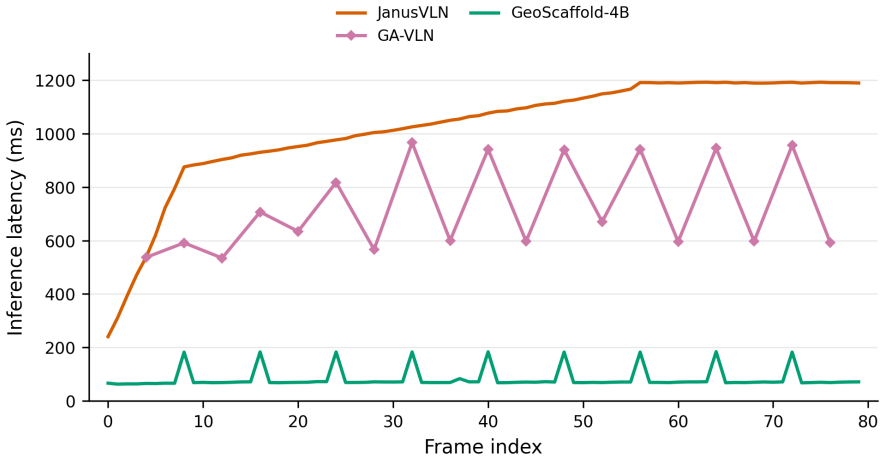

Figure 8: Per-frame inference latency over an 80-frame episode on a single RTX 4090 for GeoScaffold-4B, GA-VLN, and JanusVLN.

C Additional Implementation Details

C.1 Training Details

Besides R2R-CE and RxR-CE, the full-data policies are trained on ScaleVLN (Wang et al., 2023b), DAgger (Ross et al., 2011) trajectories, LLaVA-Video-178K (Zhang et al., 2025c), ScanQA (Azuma et al., 2022), and MMC4 (Zhu et al., 2023). The depth tokenizer is a 0.74M-parameter VQ-VAE trained offline for 20 epochs on 256^2 normalized depth maps collected along the R2R-CE and RxR-CE training trajectories. It produces $N{=}100$ tokens on a $10{\times}10$ grid over a $C{=}128$-entry codebook and stays frozen during policy training. The connectivity and traversability heads f_c and f_{tr} each predict a mask at $H_g{\times}W_g{=}256{\times}256$, so $M{=}65{,}536$ in Equation (4). Each head maps the flattened query states of its group to an $8{\times}8$ feature map with a linear layer and upsamples it to full resolution with five convolution and $2\times$ bilinear upsampling stages. Policies are trained for one epoch with AdamW at a learning rate of $2{\times}10^{-5}$, with $5{\times}10^{-6}$ for the vision tower, in bf16 under ZeRO-2. The loss weights of Equation (9) are $\lambda_{\mathrm{tok}}{=}0.25$, $\lambda_{\mathrm{rec}}{=}0.5$, and $\lambda_c{=}\lambda_{\mathrm{tr}}{=}0.25$, and the Dice term of Equation (8) uses $\lambda_{\mathrm{dice}}{=}0.25$.

C.2 Auxiliary Target Generation

Both binary masks are produced offline for every training frame with Grounding-DINO (Liu et al., 2024) and SAM (Kirillov et al., 2023) and stored next to the RGB frames, so neither model runs during policy training or at inference. Grounding-DINO is queried in open-vocabulary mode with one fixed text prompt per target and returns boxes for the matching regions. SAM then segments each box with a box prompt, and the union of the resulting segments is binarized into a single mask per frame at the RGB resolution of 640×480. The connectivity prompt names structures that connect spaces, namely door, open doorway, door opening, corridor entrance, hallway entrance, passage opening, exit doorway, stairs entrance, and stairway entrance. The traversability prompt names walkable surfaces, namely floor, ground, and stairs. Detections whose label is a whole-scene category, such as wall, ceiling, or room, are dropped, and boxes or masks covering an implausibly large fraction of the frame are discarded to suppress scene-level responses. Frames without an accepted detection receive an empty mask. During training the stored masks are downsampled to the prediction resolution $H_g \times W_g$ of the mask heads and re-binarized at 0.5.

D Qualitative Results

D.1 Attention Map Analysis at the Action-Decoding Position

We compare GeoScaffold with the same Qwen3-VL-4B policy trained without its geometric supervision, on identical inputs. Expert trajectories from the training split are replayed with the earlier actions teacher-forced, so the two rows of each example differ only in the policy. We read the head-averaged attention of the last decoder layer at the position that decodes the current action and renormalize it separately within the instruction tokens and within the visual tokens of the current dialogue window.

In these examples, GeoScaffold assigns higher attention weights to words expressing direction and spatial relations, as well as to connecting structures and nearby traversable regions. The policy without GeoScaffold shows a more diffuse visual attention distribution.

For example, Figure 9 shows two replayed turns. In the first example the expert action at the current frame is RIGHT. GeoScaffold attends more strongly to the words that locate the agent on its route, right and Enter, and places higher attention on the door and on the floor region of the current frame. In the second example the expert action is STOP. GeoScaffold attends more to the action word wait and to the relational words past and front, and its attention on the current frame is concentrated on the target door rather than spread across the scene.

Over all 768 replayed turns, GeoScaffold places more attention than the policy trained without it on the directional and geometric words of the instruction in 68% of the turns and on the connective regions of the image in 74%. Here the instruction score is the attention share on a fixed lexicon of directional and passage words such as left, past, and door, and the image score is the attention mass inside the connectivity mask of the current frame minus the mask area fraction.

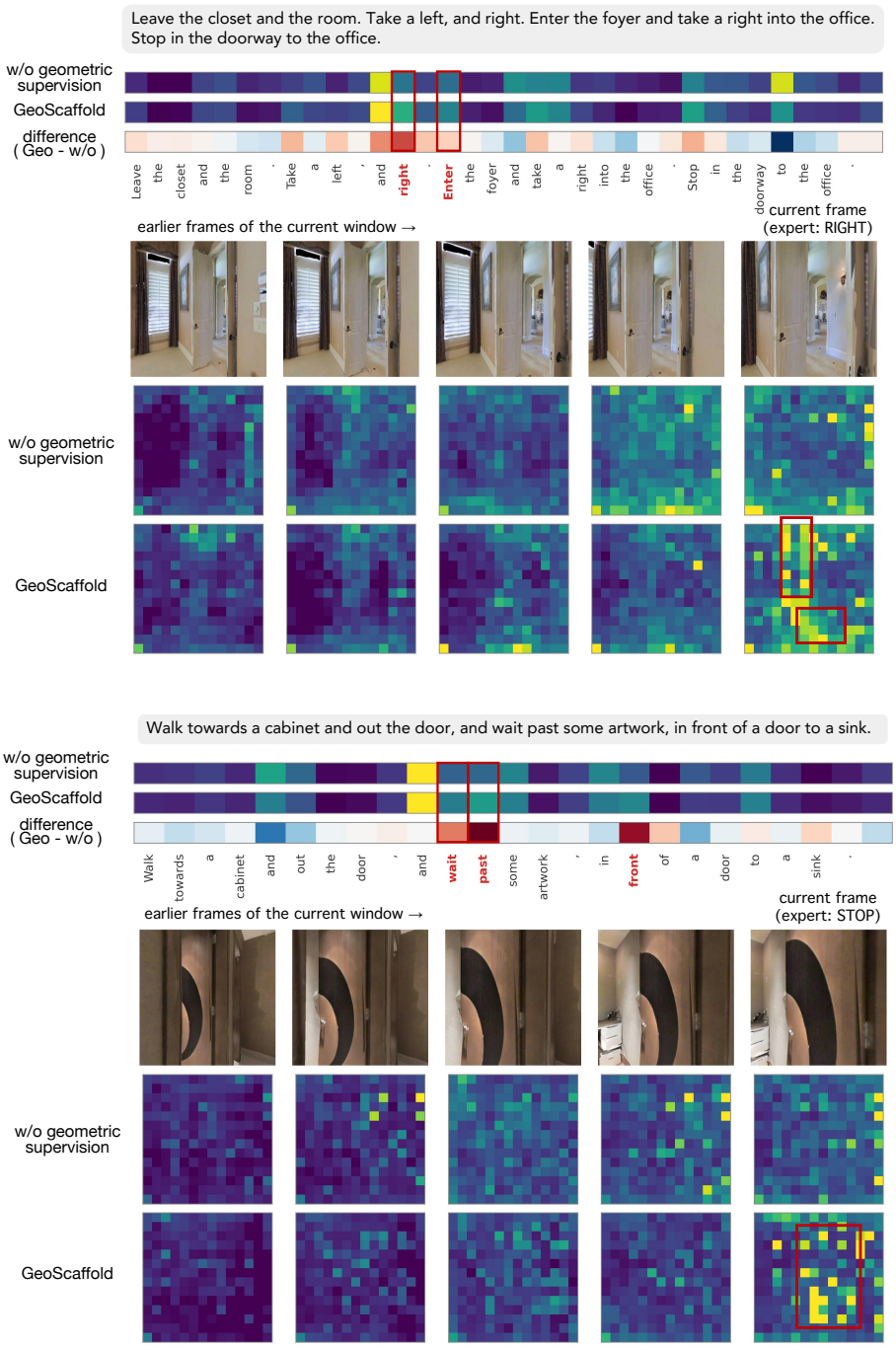

Figure 9: Attention of the action-decoding position over the instruction (strips) and over the five most recent frames of the current window (heat maps) for the same policy without and with GeoScaffold, at the last layer, averaged over heads and renormalized within each modality. Within an example the two rows share one color scale stretched to the 5th to 99th percentile, and brighter denotes higher attention. The third strip is the difference between the policies (red: GeoScaffold higher). Each cell is one token, written underneath, with the tokens discussed in the text in red. The current frame and the expert action are on the right.

D.2 More Simulated Qualitative Results

Figure 10 shows four more zero-shot simulation episodes in unseen environments. As in the main text, the policy without and with GeoScaffold starts from the same pose with the same instruction. In these episodes, the policy without GeoScaffold tends to take the wrong branch where the route splits and then drifts away from the described path, whereas GeoScaffold chooses the correct doorway, corridor, or turn at such decision points, even in complex multi-room layouts. GeoScaffold also stops closer to the goal. It recognizes when the described destination has been reached and halts there, rather than overshooting it or ending in an unrelated room. Both behaviors are consistent with the spatial structure that GeoScaffold internalizes during training.

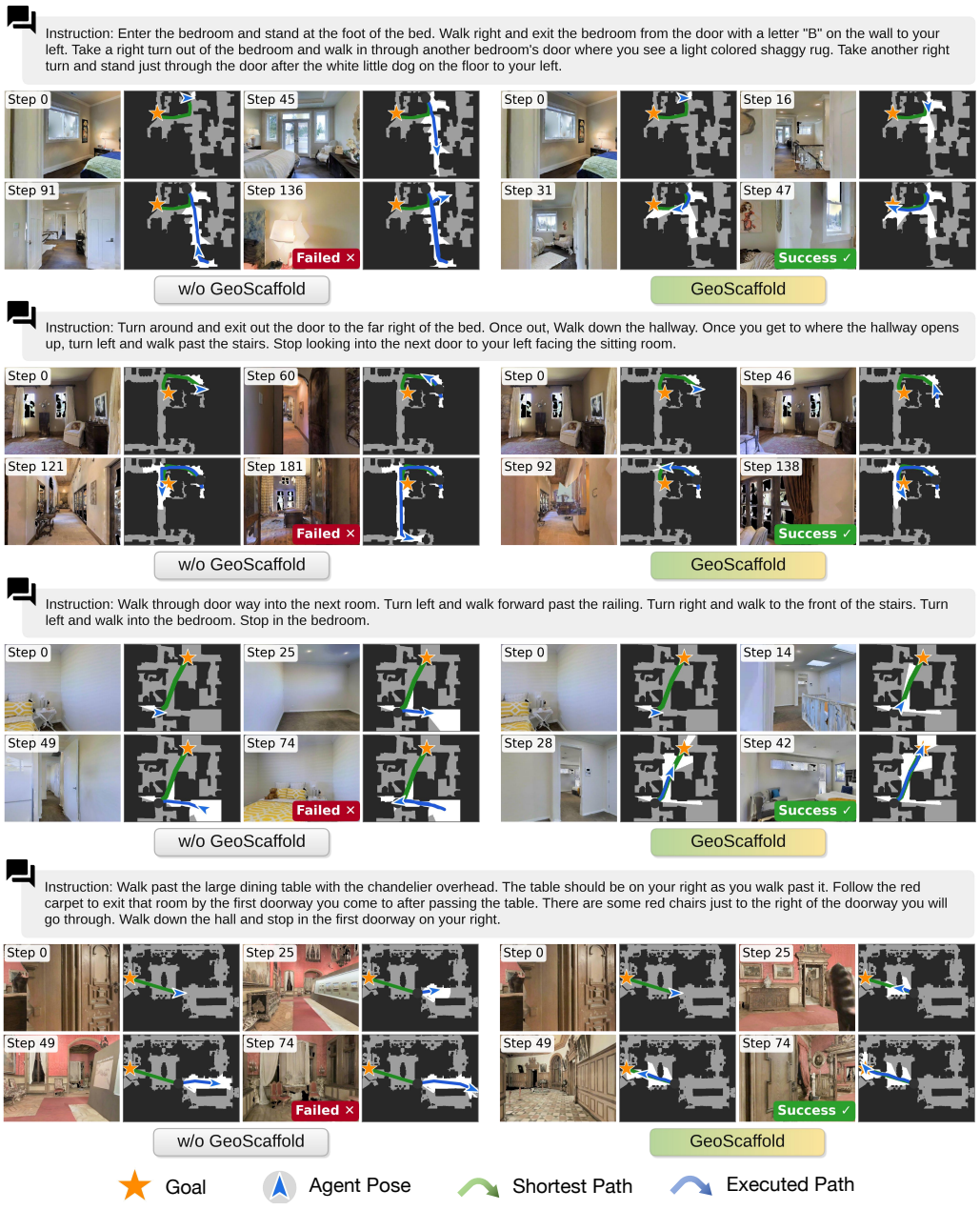

Figure 10: More zero-shot simulated qualitative comparisons for the same policy without and with GeoScaffold.

D.3 More Real-World Qualitative Results

Figure 11 shows more zero-shot episodes of the Qwen3-VL-4B policy with GeoScaffold on the Unitree Go2 Edu, which again uses only the RGB stream of its RealSense D435i camera. Across our real-world experiments, the policy handles a broad range of navigation behaviors, including following paths and roads, avoiding obstacles that partially block the route, traversing long corridors with turns and doorways, and stopping at described landmarks such as a sofa, a door, or a pillar. It completes these tasks with high success rates indoors and improves over the policy without GeoScaffold in every setting (Figure 7). These results show that the depth-aware spatial reasoning acquired in simulation transfers to unseen real-world scenes and generalizes across indoor and outdoor environments.

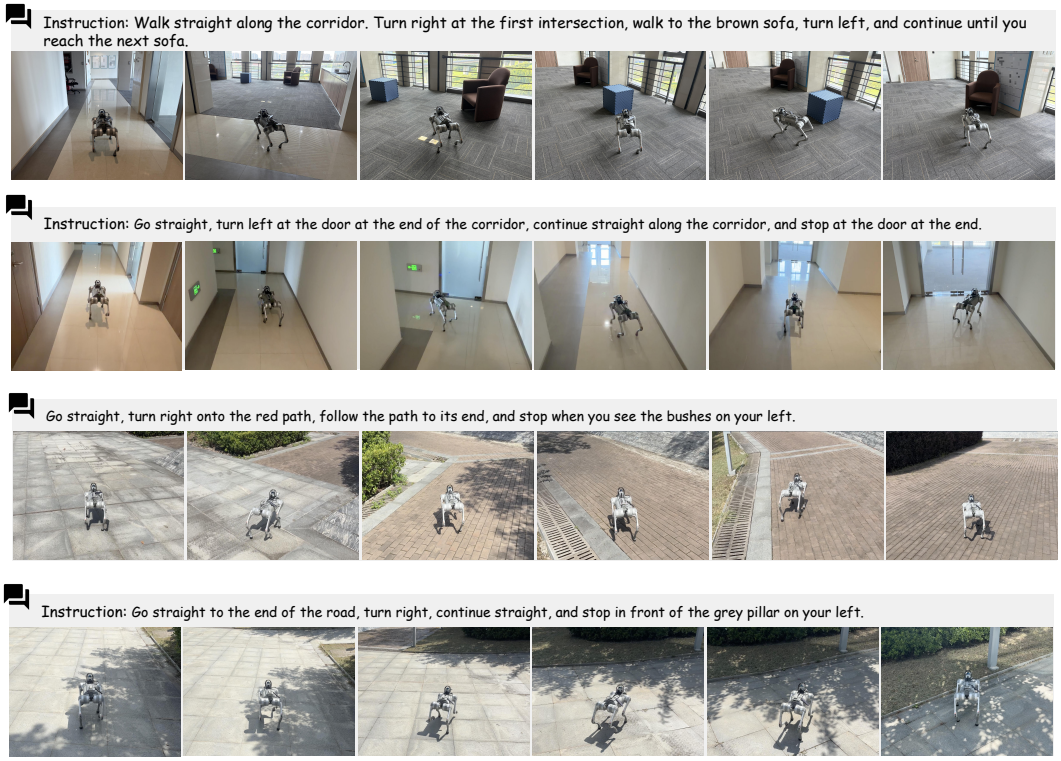

Figure 11: More zero-shot real-world qualitative results of GeoScaffold.