PANOVLN: TOWARDS EFFECTIVE PANORAMIC VISION-AND-LANGUAGE NAVIGATION

Anonymous authors
Paper under double-blind review

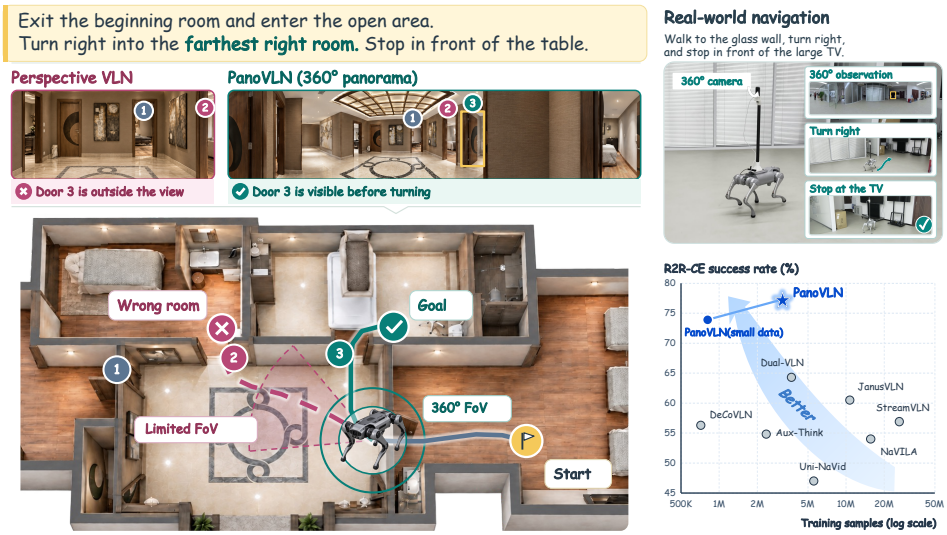

Figure 1: PanoVLN fully exploits the wider visual context of panoramas to make vision-and-language navigation more accurate and efficient.

ABSTRACT

Recent vision-language models (VLMs) have advanced vision-and-language navigation (VLN), enabling models to predict navigation actions from visual observations and language instructions. In this work, we explore VLN with panoramic observations and introduce **PanoVLN**. The motivation is straightforward: more complete visual context should enable better-informed navigation decisions. For example, a panorama can reveal a passage outside a perspective camera's field of view, allowing the model to identify the intended route without additional exploration. However, we find that simply replacing perspective images with panoramas yields only limited gains. Our diagnosis suggests that fully exploiting wider visibility requires modifications to action prediction, training supervision, and visual representation. First, wider visibility supports longer-horizon action planning. We make the model predict longer action sequences, enabling larger turns and subsequent movement from a single panorama. Specifically, we introduce a confidence-guided execution (CGE) strategy that dynamically determines how many predicted actions to execute before replanning. Second, wider visibility also brings more complex route choices. We therefore construct training routes with frequent branching points and clear instructions to provide targeted supervision for route selection. Third, panoramic navigation requires understanding spatial relationships across viewing directions, beyond recognizing individual landmarks. We combine semantic and geometric features from RGB panoramas to capture both scene content and spatial layout without adding visual tokens. With a 4B backbone and RGB-only input, PanoVLN surpasses the previous SOTA by 11.9% and 8.7% in success rate on R2R-CE and RxR-CE Val-Unseen. Real-world experiments on a quadruped further demonstrate faster navigation with fewer pauses than prior VLN methods.

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.

1 INTRODUCTION

Vision-and-language navigation (VLN) requires an agent to navigate through an environment by following a natural-language instruction (Anderson et al., 2018; Krantz et al., 2020). Recent vision-language models (VLMs) have advanced this task by predicting navigation actions from visual observations and instructions (Zhang et al., 2024; Cheng et al., 2025; Wei et al., 2026b).

Most of the previous works take perspective images as input; in this work, we explore whether panoramic observations can further improve navigation by making more of the surrounding environment available for each decision. The motivation is straightforward: an equirectangular panorama (ERP) provides a 360° view, exposing passages, landmarks, and route alternatives across different viewing directions and providing a more complete visual basis for navigation decisions.

However, we find that simply replacing perspective images with ERPs under the same training and inference setup does not improve performance. Motivated by this observation, we conduct a detailed diagnostic analysis and find that exploiting panoramic context requires targeted adaptations. Specifically, we adapt action prediction, training supervision, and visual representation to panoramic inputs, and propose PanoVLN.

First, as a route changes direction, it may extend beyond the left or right edge of a perspective image. A panorama's 360° view can show where the route continues, providing visual context for a longer sequence of navigation actions. We therefore train PanoVLN with a longer prediction horizon, and our horizon study shows that panoramic policies favor much longer action sequences than perspective policies. Predictions farther into the future are naturally less reliable, so executing the entire sequence is not always desirable. We therefore introduce *confidence-guided execution* (CGE), which uses action uncertainty to determine how many predicted actions to execute before reobserving and replanning.

Second, a panorama is particularly useful at a branching point, where its wider field of view can reveal several possible paths. However, branching points are relatively sparse in existing VLN training data, providing limited supervision for learning to use this advantage. We therefore construct a dataset of 98K trajectories across 800 HM3D scenes (Ramakrishnan et al., 2021), with routes that contain frequent branching points and instructions that clearly specify which path to take. We generate and verify the instructions against route observations to ensure that the described movements and choices are visually grounded. We further increase the sampling frequency around turns and stopping points, providing stronger supervision where route decisions and completion matter most. Together, these choices provide dense, targeted supervision for learning route decisions and following the selected path to completion.

Third, using panoramic observations for navigation requires understanding the spatial relationships among different parts of the scene. The VLM's visual features mainly capture scene semantics, while navigation also depends on how landmarks, passages, and other scene elements are spatially arranged. We therefore combine the VLM's semantic features with geometric features extracted by PanoVGGT (Guo et al., 2026b) from the same RGB panorama. We align and fuse features from corresponding ERP regions, providing both semantic and geometric information without additional visual tokens or depth input.

With a 4B backbone and RGB-only observations, PanoVLN achieves success rates of 77.3% on R2R-CE Val-Unseen and 78.0% on RxR-CE Val-Unseen, exceeding the previous state of the art by 11.9% and 8.7%, respectively. On a quadruped robot, PanoVLN navigates indoor and outdoor routes in less time, with fewer policy calls and pauses than prior VLN baselines.

2 RELATED WORK

Vision-and-language navigation. Vision-and-language navigation in continuous environments (VLN-CE) extends instruction-following routes from navigation graphs to executable motion (Anderson et al., 2018; Ku et al., 2020; Krantz et al., 2020). Waypoint and map-based methods predict reachable locations, maintain spatial representations, or look ahead along candidate routes to support planning and control (Hong et al., 2022; Georgakis et al., 2022; Wang et al., 2023a; An et al., 2025; Wang et al., 2024). VLM-based policies use visual histories or streaming video to predict navigation commands, sometimes passing intermediate decisions to a separate execution policy (Zhang et al., 2024; 2025a; Wei et al., 2026b; Cheng et al., 2025; Wei et al., 2026a). Complementary work improves

waypoint supervision, the scale of navigation data, and instruction generation (Raychaudhuri et al., 2021; Wang et al., 2023b; Yan et al., 2024). PanoVLN builds on these advances to study how full-surround visual observations support vision-and-language navigation.

Panoramic geometry. Panoramic geometry methods account for spherical projection when estimating depth and 3D scene structure. Depth estimation has used ERP–cubemap fusion, camera-independent spherical representations, and models trained for panoramic inputs (Jiang et al., 2021; Piccinelli et al., 2025; Li et al., 2026; Lin et al., 2025). Feed-forward reconstruction models jointly estimate camera and scene geometry from images, with PanoVGGT extending this approach to panoramas (Wang et al., 2025a; Guo et al., 2026b). Navigation methods likewise represent spatial layout through cross-modal maps, grid memories, lookahead scene features, or separate spatial and semantic memories (Georgakis et al., 2022; Wang et al., 2023a; 2024; Zeng et al., 2026). PanoVLN brings panoramic geometry into VLN to strengthen spatial reasoning.

3 METHOD

In this work, we study how to fully exploit the complete visual context provided by panoramas in VLN task. Starting from a baseline model with panoramas as input, we make three key adaptations: First, to fully utilize the wider visibility, we train and execute in longer action sequences. Second, we construct a decision-centric training dataset more suitable for panoramic settings. Third, we develop geometric-aware visual representations to better understand the panoramic observations.

3.1 PRELIMINARIES

Perspective baseline. We begin with a VLM-based policy for VLN-CE (Krantz et al., 2020). At step t, the policy receives a natural-language instruction x, the current perspective RGB image I_t, and sampled visual history $\mathcal{I}_{<t}$. A visual encoder extracts image features, which are projected into the language model's embedding space and combined with instruction tokens. The language model autoregressively predicts a sequence of actions from $\mathcal{A} = \{\texttt{forward}, \texttt{left}, \texttt{right}, \texttt{stop}\}$. The policy learns from expert action sequences and, at inference, executes a fixed-length prefix of its prediction before observing again. Movement actions use fixed translation and rotation increments. The `stop` action ends the episode, and success requires stopping within the benchmark's goal region.

Panoramic baseline. We obtain a panoramic baseline by replacing both current and historical perspective images with RGB equirectangular panoramas (ERPs) during training and inference. An ERP linearly maps a $360^\circ \times 180^\circ$ field of view to a rectangle. The agent's heading is centered, and the left and right image boundaries are adjacent across a seam behind the agent. The policy input becomes $\mathcal{O}_t = (x, \mathcal{P}_{<t}, P_t)$, where P_t is the current ERP and $\mathcal{P}_{<t}$ is the sampled panoramic history. We retain the same VLM, training trajectories, action supervision, and execution procedure. In our experiments, this input-only replacement yields limited gains and can even reduce navigation performance, motivating the adaptations described below.

3.2 LONGER ACTION-SEQUENCE SUPERVISION AND EXECUTION

A panorama can show where a route continues after it turns beyond the field of view of a perspective image. Short action targets use only part of this visual context for supervision. We therefore use longer action-sequence supervision and adapt the execution length according to prediction uncertainty.

Action-sequence supervision. At training state t, the target $\mathbf{A}_t^* = (a_{t,1}^*, \ldots, a_{t,H}^*)$ contains the next H expert actions, padded with `stop` beyond the trajectory end. The policy predicts these actions autoregressively. Using teacher forcing, we minimize

$$\mathcal{L}_{\text{act}} = -\frac{1}{H} \sum_{i=1}^{H} \log p_\theta \left(a_{t,i}^* \mid \mathcal{O}_t, \mathbf{A}_{t,<i}^* \right), \tag{1}$$

where $\mathbf{A}_{t,<i}^*$ contains the preceding expert actions and p_θ is the VLM's next-token distribution.

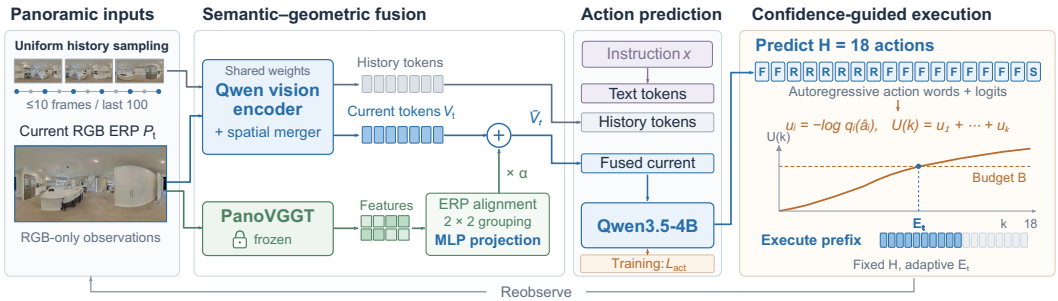

Figure 2: **PanoVLN pipeline.** Current and historical panoramas share a fixed visual-token budget. Aligned geometric features enrich the current visual tokens, and uncertainty determines how much of the predicted action sequence to execute.

Confidence-guided execution (CGE). The prediction horizon H determines how far ahead the policy predicts, whereas the execution length determines when it reobserves. CGE selects this length according to the uncertainty of the predicted actions.

Given $\mathcal{O}_t$ and preceding predictions, let $z_{t,i}(a)$ be the logit for action a at position i. Normalizing over $\mathcal{A}$, we define uncertainty for the generated action $\hat{a}_{t,i}$ as

$$q_{t,i}(a) = \frac{\exp z_{t,i}(a)}{\sum_{b\in\mathcal{A}} \exp z_{t,i}(b)}, \qquad u_{t,i} = -\log q_{t,i}(\hat{a}_{t,i}). \tag{2}$$

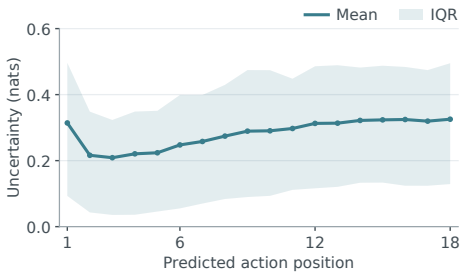

Figure 3: **Action uncertainty.**Uncertainty rises after an initial dip and varies across policy calls, motivating adaptive execution.

Mean uncertainty rises after an initial dip, with substantial variation across policy calls (Figure 3).

Let $U_t(k) = \sum_{i=1}^{k} u_{t,i}$, with $U_t(0) = 0$. CGE extends the prefix while $U_t(k) \le B$ for an uncertainty budget B, selecting at least $E_{\min}$ actions:

$$E_t = \max\{k \in \{1,\dots,H\} : k \le E_{\min} \text{ or } U_t(k) \le B\}. \tag{3}$$

The agent executes this prefix, then reobserves unless it stops.

3.3 Decision-Centric Data Construction

Existing VLN training data contains relatively few trajectories with frequent route choices among multiple visible paths. This provides limited supervision for learning to select the intended path from panoramic observations. We therefore construct trajectories with frequent branching points and pair them with instructions that clearly identify the chosen path. We further sample turns and stopping points more densely during data construction.

Route construction and filtering. We construct 98K navigation trajectories across 800 HM3D scenes, dividing walkable space into connected areas using the navigation mesh. A *branching point* has at least two visible, traversable paths to different areas, excluding the incoming path. We sample endpoints in different areas and retain routes through branching points. Rendering-quality checks remove candidates with mesh holes or incomplete geometry, followed by near-duplicate removal. An expert converts the remaining routes into primitive action sequences; replay verifies goal reachability and visibility of the chosen path and its alternatives at each branching point.

Instruction construction. We divide trajectories by route events into *travel*, *branching*, and *arrival* segments. Each segment uses first-person video with the expert path marked on the ground; branching and arrival also use eight-view compass images. Qwen3.8-27B (Qwen Team, 2026b) describes movement, identifies the chosen path from visible cues, and specifies the stopping location. We combine descriptions in route order, remove repetition, and refine the wording.



Figure 4: **Data construction pipeline**. We construct routes with frequent branching points, generate and verify instructions, and sample training states.

We verify each instruction segment using clean videos and compass images without instruction or route overlays. Motion consistency checks movement order and turn directions against the replay. Choice grounding checks whether the instruction identifies the demonstrated entrance among alternatives; stop grounding checks whether it describes the observed arrival area. Mismatched segments are revised locally and reverified; only samples passing all three checks are retained.

Training sample selection. Adjacent states often have similar observations and overlapping action targets. For $H = 18$, we use a stride-six grid, reducing overlap between adjacent grid targets from 17 to 12 actions. We add states at sustained-turn onsets and near termination to supervise turning and stopping. Each state is paired with its H-step expert action sequence. The grid preserves route coverage; added states emphasize action transitions.

3.4 Geometry-aware visual representation

A panoramic observation brings different parts of the surrounding scene into a single view, making their spatial relationships important for navigation. We therefore fuse semantic and aligned geometric features from the current RGB panorama without adding visual tokens.

Visual context allocation. We allocate N_c tokens to the current ERP and $N_h < N_c$ tokens to each history frame. The finer current features support route selection, while coarser historical features provide context for instruction progress. We uniformly sample up to M past observations from a recent temporal window, giving a total visual-token budget of $N_c + MN_h$.

Spatially aligned geometric fusion. A pretrained PanoVGGT encoder (Guo et al., 2026b) extracts geometric features from the current RGB panorama. We resample them in ERP coordinates and group them to cover the same regions as the VLM's merged current tokens V_t. A trainable MLP f_ψ projects the aligned geometric groups G_t into the visual-token embedding space for residual fusion:

$$\bar{V}_t = V_t + \alpha f_\psi(G_t), \tag{4}$$

where α is a fixed residual scale. Fusion combines semantics and geometry from corresponding ERP regions, preserving token count and order. The instruction, history, and fused tokens condition action prediction. We train the VLM and projection with Eq. 1 and freeze the geometry encoder.

Table 1: **Simulation benchmark comparison.** We evaluate performance on R2R-CE and RxR-CE Val-Unseen with the indicated observation modalities. * denotes the waypoint predictor from Hong et al. (2022); † denotes training without navigation data beyond R2R-CE and RxR-CE. NE is in meters; other scores are percentages. PanoVLN achieves the highest SR and SPL on both benchmarks.

Method	Observation				R2R Val-Unseen				RxR 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
GridMM* Wang et al. (2023a)	✓	✓	✓		5.11	61.0	49.0	41.0	–	–	–	–
ETPNav* An et al. (2025)	✓	✓	✓		4.71	65.0	57.0	49.0	5.64	54.7	44.8	61.9
HNR* Wang et al. (2024)	✓	✓	✓		4.42	67.0	61.0	51.0	5.50	56.3	46.7	63.5
ScaleVLN* Wang et al. (2023b)	✓	✓	✓		4.80	–	55.0	51.0	–	–	–	–
InstructNav Long et al. (2025)	✓	✓	✓		6.89	–	31.0	24.0	–	–	–	–
LAW Raychaudhuri et al. (2021)		✓	✓	✓	6.83	44.0	35.0	31.0	10.90	8.0	8.0	38.0
CM^2 Georgakis et al. (2022)		✓	✓	✓	7.02	41.5	34.3	27.6	–	–	–	–
WS-MGMap Chen et al. (2022)		✓	✓	✓	6.28	47.6	38.9	34.3	–	–	–	–
CMA Krantz et al. (2020)			✓	✓	7.37	40.0	32.0	30.0	–	–	–	–
MapNav† Zhang et al. (2025b)		✓	✓	✓	4.93	53.0	39.7	37.2	7.62	32.6	27.7	43.5
StreamVLN† Wei et al. (2026b)				✓	5.43	62.5	52.8	47.2	6.72	48.6	42.5	60.2
Aux-Think† Wang et al. (2025b)				✓	6.08	60.0	54.8	46.9	6.24	52.2	40.2	–
JanusVLN† Zeng et al. (2026)				✓	5.17	58.0	52.8	49.2	6.46	51.4	44.3	59.1
CorrectNav† Yu et al. (2026)				✓	4.24	67.5	65.1	62.3	4.09	69.3	63.3	**75.2**
PanoVLN† (Ours)	✓				3.10	79.7	73.9	67.9	3.14	74.1	63.9	71.4
NaVid Zhang et al. (2024)				✓	5.47	49.1	37.4	35.9	–	–	–	–
Uni-NaVid Zhang et al. (2025a)				✓	5.58	53.3	47.0	42.7	6.24	48.7	40.9	–
NaVILA Cheng et al. (2025)				✓	5.22	62.5	54.0	49.0	6.77	49.3	44.0	58.8
StreamVLN Wei et al. (2026b)				✓	4.90	63.6	56.4	50.2	5.65	54.4	45.4	63.7
JanusVLN Zeng et al. (2026)				✓	4.78	65.2	60.5	56.8	6.06	56.2	47.5	62.1
DualVLN Wei et al. (2026a)				✓	4.05	70.7	64.3	58.5	4.58	61.4	51.8	70.0
AwareVLN Guo et al. (2026a)				✓	4.02	73.5	65.4	55.1	3.95	67.6	56.1	65.7
PanoVLN (Ours)	✓				**2.83**	**83.1**	**77.3**	**70.6**	**2.85**	**78.0**	**65.9**	73.3

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

We evaluate on the R2R-CE and RxR-CE Val-Unseen splits (Krantz et al., 2020; Ku et al., 2020) in Matterport3D scenes (Chang et al., 2017) using Habitat (Savva et al., 2019). We report navigation error (NE), oracle success rate (OS), success rate (SR), success weighted by path length (SPL), and normalized dynamic time warping (nDTW). OS records whether a trajectory comes within $3\,\mathrm{m}$ of the goal; SR also requires stopping there. SPL measures path efficiency and nDTW reference-route agreement. NE is in meters; other scores use a 0–100 scale.

PanoVLN uses RGB-only observations. The default model combines Qwen3.5-4B (Qwen Team, 2026a) with a frozen PanoVGGT encoder (Guo et al., 2026b) and uses an $H = 18$ prediction horizon with CGE for execution. PanoVLN† trains on R2R-CE and RxR-CE navigation data; the full model additionally uses our constructed dataset. Ablation configurations are specified with each study. Training settings and the navigation prompt are in Appendix B.

4.2 SIMULATION EXPERIMENTS

Comparison with prior methods. PanoVLN achieves state-of-the-art SR of 77.3% on R2R-CE and 78.0% on RxR-CE, exceeding the previous best results by 11.9 and 8.7 percentage points, respectively (Table 1). PanoVLN† also leads the restricted-data group in SR and SPL on both benchmarks. Adding our decision-centric trajectories further improves performance in unseen scenes.

Use of panoramic context. With the same ERP and instruction, short-horizon training concentrates attention ahead, resembling a perspective policy, while PanoVLN attends to instructed passages across directions. Short targets often share initial movements across routes; longer targets include



Figure 5: **Real-world navigation.** PanoVLN deployed on a quadruped robot and transfers to real-world instruction following without scene-specific fine-tuning.

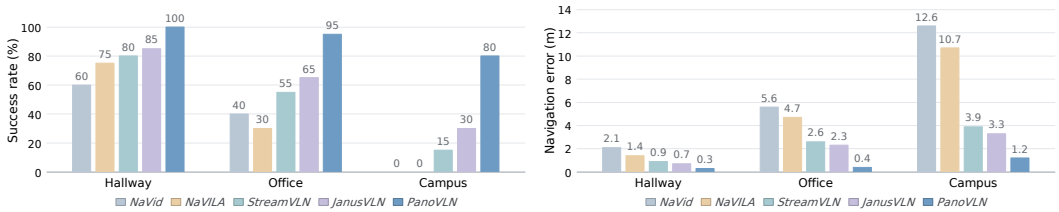

Figure 6: **Real-world navigation performance.** We compare SR and NE on 20 shared instruction–route pairs each in Hallway, Office, and Campus and achieves the best navigation performance.

route choices and subsequent movement, making the distinguishing visual cues relevant to prediction and encouraging use of the full panorama.

4.3 Real-World Experiments

Deployment and evaluation. All methods use a Unitree Go2, the same Insta360 X5 mounted 1.5 m above ground, and a remote RTX 3090. PanoVLN uses approximately 12 GB of GPU memory; network overhead averages 272 ms per call, excluding inference. Execution is synchronous: the robot waits for a response, executes its actions, then requests the next prediction.

We compare NaVid (Zhang et al., 2024), NaVILA (Cheng et al., 2025), StreamVLN (Wei et al., 2026b), JanusVLN (Zeng et al., 2026), and PanoVLN on 20 shared instruction–route pairs per setting, without scene-specific fine-tuning. Hallway tests successive turns; Office adds clutter and room transitions. Campus covers gardens, sports fields, and plazas, testing transfer from indoor training to outdoor spaces and varied terrain.

Navigation performance. The results test both local route complexity and transfer across scene types (Figure 6; qualitative trials are in the supplementary video). NaVid struggles with instruction progress over long routes, while clutter and room transitions disrupt NaVILA in Office. StreamVLN and JanusVLN handle indoor routes more reliably but struggle to transfer to open outdoor layouts in Campus. PanoVLN follows successive indoor route choices and transfers this ability outdoors, maintaining instruction-to-route grounding across changes in appearance and spatial layout.

Execution efficiency. Table 2 reports trial averages (see Appendix B.2). *Time* is navigation duration; *Speed* is traveled distance divided by duration, including waiting. *Wait* is the fraction of time awaiting policy responses; *Pauses* counts stationary intervals longer than 1 s. *Calls* counts policy requests; *Latency* is inference time per request, excluding network communication.

Under synchronous execution, frequent policy requests add inference and communication delay and interrupt motion. JanusVLN requests a prediction for each action; StreamVLN has lower per-call latency but replans every four actions. PanoVLN predicts longer segments, and CGE selects a

Table 2: **Real-world execution efficiency.** PanoVLN achieves the best overall navigation efficiency among the compared methods.

Method	Navigation		Continuity		Planning	
	Time (s) ↓	Speed (cm/s) ↑	Wait (%) ↓	Pauses ↓	Calls ↓	Latency (s) ↓
NaVid	135.2	12.9	23.2	15.7	29.4	0.90
NaVILA	194.0	8.2	24.7	29.6	41.3	1.05
StreamVLN	117.8	13.3	27.1	8.3	34.3	**0.58**
JanusVLN	412.9	5.3	39.2	95.3	101.7	1.32
PanoVLN (Ours)	**86.4**	**25.7**	**13.6**	**5.4**	**7.4**	1.08

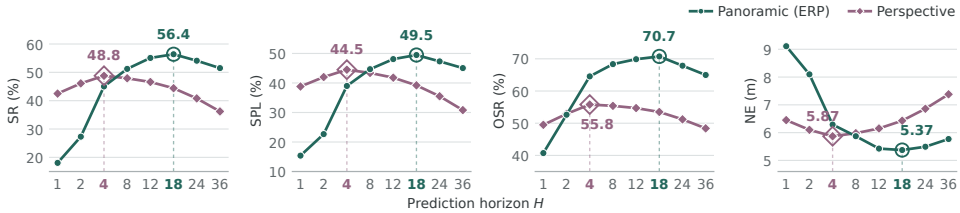

Figure 7: **Prediction horizon ablation.** Panoramic policies benefit from longer supervision and outperform perspective policies at longer horizons.

confident prefix before requesting a new observation. Sustaining motion while predictions remain confident reduces interruptions and total navigation time.

4.4 Ablation Studies

The horizon, execution, sampling, and geometry studies use R2R-CE and RxR-CE training data; the data study varies the additional trajectories. Evaluation uses the corresponding Val-Unseen splits.

Prediction horizon. We vary the prediction horizon while fixing the VLM, visual-token budget, and random-start sampling, training geometry-free policies for 4,000 updates (Figure 7). ERP policies trail perspective policies at short horizons but overtake them as targets lengthen. Perspective performance peaks at $H = 4$, while ERP favors $H = 18$.

Short targets may end before visible routes diverge; longer ERP targets supervise the intended choice. Longer perspective targets increasingly require cues outside the current view. ERP improves from $H = 8$ to $H = 18$ with execution fixed at six actions, linking the gain to longer supervision. These results favor matching supervision to visible route information; we use $H = 18$ thereafter.

Training-state sampling. We compare random starts with our turn- and termination-aware sampling. Both variants use a geometry-free $H = 18$ architecture, 4,000 updates, and six-action execution. The random variant is the $H = 18$ ERP run in Figure 7.

Table 3: **Training data comparison.**

Sampling	NE↓	OS↑	SR↑	SPL↑
Random	5.37	**70.7**	56.4	49.5
Ours	**4.40**	69.9	**62.8**	**57.2**

Long stretches of forward motion supply similar targets, while brief turn and stop states determine route transitions and completion. Our sampling gives these states more supervision (Table 3). Comparable OS and higher SR indicate more reliable termination in the goal region, highlighting the value of learning when to change or end an action sequence.

Confidence-guided execution. Table 4 compares fixed, random, and confidence-guided execution using the same geometry-free $H = 18$ checkpoint, trained for one epoch with our sampling.

Long fixed prefixes commit to increasingly uncertain actions, while random prefixes ignore the model's confidence. CGE adapts execution to each state, outperforming random prefixes and all fixed strategies, including one-action execution. These studies separate two choices: long targets teach route selection, while CGE determines when to reobserve.

Table 4: **Execution strategy comparison.** We compare fixed, random, and confidence-guided execution. CGE achieves the best navigation performance on both benchmarks.

Execution strategy	R2R-CE				RxR-CE			
	NE↓	OS↑	SR↑	SPL↑	NE↓	SR↑	SPL↑	nDTW↑
Fixed: 1 action	4.42	68.2	64.6	60.5	4.11	65.7	55.6	70.3
Fixed: 6 actions	4.43	70.8	64.2	58.7	4.25	65.4	54.2	69.0
Fixed: 12 actions	4.65	69.7	61.4	55.4	4.98	61.1	46.4	64.6
Fixed: 18 actions	4.89	67.0	57.2	50.9	5.64	54.9	43.4	59.8
Random: 1–18 actions	4.62	68.5	61.3	55.6	5.07	59.9	52.3	64.9
CGE (Ours)	**4.10**	**71.0**	**66.6**	**61.5**	**4.06**	**66.5**	**56.4**	**70.8**

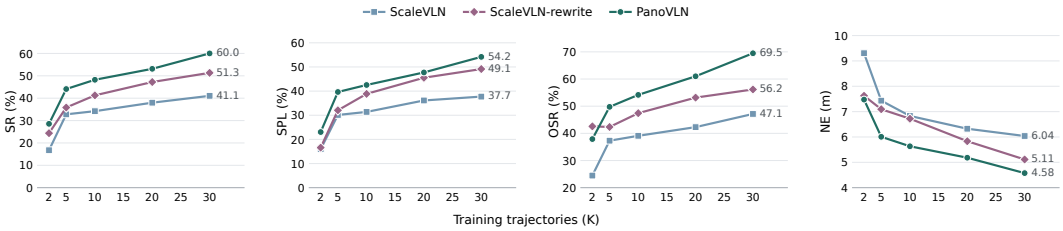

Figure 8: **Training data comparison.** We compare data sources across training scales on R2R-CE Val-Unseen.Our data consistently delivers the best navigation performance.

Table 5: **Geometry encoder comparison.** We compare geometric features from different encoders. PanoVGGT yields the highest success rates on both benchmarks.

Extra encoder	R2R-CE				RxR-CE			
	NE↓	OS↑	SR↑	SPL↑	NE↓	SR↑	SPL↑	nDTW↑
None	4.10	71.0	66.6	61.5	4.06	66.5	56.4	70.8
UniK3D	**3.77**	73.2	67.8	63.3	**3.92**	65.4	**57.3**	68.1
DA^2	4.01	72.3	65.5	61.6	4.35	63.0	55.9	67.7
DAP	4.15	70.9	64.5	60.4	4.48	61.1	54.3	66.5
PanoVGGT (Ours)	3.91	**73.6**	**68.6**	**63.7**	4.02	**67.1**	57.2	**71.5**

Geometric features. We compare frozen UniK3D (Piccinelli et al., 2025), DA^2 (Li et al., 2026), DAP (Lin et al., 2025), and PanoVGGT encoders with the same fusion design, visual-token budget, and CGE (Table 5). The alternative encoders have mixed effects, whereas PanoVGGT improves SR on both benchmarks. Its panoramic features link landmarks to neighboring passages and directions, complementing semantic appearance with spatial cues for route selection.

Training data. We compare ScaleVLN (Wang et al., 2023b), ScaleVLN-rewrite, and our data at matched trajectory counts, using PanoVGGT and CGE without DAgger (Ross et al., 2011) (Figure 8). ScaleVLN-rewrite retains the original routes and regenerates instructions using our instruction pipeline. Its gains over ScaleVLN demonstrate the benefit of improved instruction quality. Our full pipeline yields further gains across training scales by combining clear instructions with decision-rich routes, providing repeated supervision for language-guided selection among paths visible in a panorama. Appendices C.1 and C.2 provide data-composition and backbone comparisons.

5 Conclusion

We presented PanoVLN for vision-and-language navigation from RGB panoramas, combining longer action supervision, confidence-guided execution, decision-centric training, and geometric features. With a 4B backbone, it achieves state-of-the-art success rates on R2R-CE and RxR-CE and enables faster real-world navigation with fewer pauses. Our findings suggest that broader visibility should shape how navigation policies learn and act.

REFERENCES

Dong An, Hanqing Wang, Wenguan Wang, Zun Wang, Yan Huang, Keji He, and Liang Wang. ETPNav: Evolving topological planning for vision-language navigation in continuous environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(7):5130–5145, July 2025. doi: 10.1109/TPAMI.2024.3386695. URL https://doi.org/10.1109/TPAMI.2024.3386695.

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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3674–3683, June 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Anderson_Vision-and-Language_Navigation_Interpreting_CVPR_2018_paper.html.

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, 2025a. URL https://arxiv.org/abs/2511.21631.

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, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report, 2025b. URL https://arxiv.org/abs/2502.13923.

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 *2017 International Conference on 3D Vision (3DV)*, 2017. URL https://niessner.github.io/Matterport/.

Peihao Chen, Dongyu Ji, Kunyang Lin, Runhao Zeng, Thomas H. Li, Mingkui Tan, and Chuang Gan. Weakly-supervised multi-granularity map learning for vision-and-language navigation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 38149–38161, 2022. doi: 10.52202/068431-2764. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/f959b05dd74ba8a735276c3df4ae8b71-Abstract-Conference.html.

An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Biyik, Hongxu Yin, Sifei Liu, and Xiaolong Wang. NaVILA: Legged robot vision-language-action model for navigation. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.018. URL https://www.roboticsproceedings.org/rss21/p018.html.

Georgios Georgakis, Karl Schmeckpeper, Karan Wanchoo, Soham Dan, Eleni Miltsakaki, Dan Roth, and Kostas Daniilidis. Cross-modal map learning for vision and language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15460–15470, June 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Georgakis_Cross-Modal_Map_Learning_for_Vision_and_Language_Navigation_CVPR_2022_paper.html.

Wenxuan Guo, Xiuwei Xu, Yichen Liu, Xiangyu Li, Hang Yin, Huangxing Chen, Wenzhao Zheng, Jianjiang Feng, Jie Zhou, and Jiwen Lu. AwareVLN: Reasoning with self-awareness for vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4065–4075, June 2026a. URL https://openaccess.thecvf.com/content/CVPR2026/html/Guo_AwareVLN_Reasoning_with_Self-awareness_for_Vision-Language_Navigation_CVPR_2026_paper.html.

Yijing Guo, Mengjun Chao, Luo Wang, Tianyang Zhao, Haizhao Dai, Yingliang Zhang, Jingyi Yu, and Yujiao Shi. PanoVGGT: Feed-forward 3D reconstruction from panoramic imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 36444–36453, June 2026b. URL https://openaccess.thecvf.com/content/CVPR2026/html/Guo_PanoVGGT_Feed-Forward_3D_Reconstruction_from_Panoramic_Imagery_CVPR_2026_paper.html.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15439–15449, June 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Hong_Bridging_the_Gap_Between_Learning_in_Discrete_and_Continuous_Environments_CVPR_2022_paper.html.

Hualie Jiang, Zhe Sheng, Siyu Zhu, Zilong Dong, and Rui Huang. UniFuse: Unidirectional fusion for 360° panorama depth estimation. *IEEE Robotics and Automation Letters*, 2021. doi: 10.1109/LRA.2021.3058957. URL https://arxiv.org/abs/2102.03550.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *Computer Vision – ECCV 2020*, volume 12373 of *Lecture Notes in Computer Science*, pp. 104–120. Springer, 2020. doi: 10.1007/978-3-030-58604-1_7. URL https://doi.org/10.1007/978-3-030-58604-1_7.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4392–4412, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.356. URL https://aclanthology.org/2020.emnlp-main.356/.

Haodong Li, Wangguandong Zheng, Jing He, Yuhao Liu, Xin Lin, Xin Yang, Ying-Cong Chen, and Chunchao Guo. DA^2: Depth anything in any direction. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/786e39208b37bfcb0ad89413f155df99-Abstract-Conference.html.

Xin Lin, Meixi Song, Dizhe Zhang, Wenxuan Lu, Haodong Li, Bo Du, Ming-Hsuan Yang, Truong Nguyen, and Lu Qi. Depth any panoramas: A foundation model for panoramic depth estimation, 2025. URL https://arxiv.org/abs/2512.16913.

Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. InstructNav: Zero-shot system for generic instruction navigation in unexplored environment. In *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2049–2060. PMLR, 2025. URL https://proceedings.mlr.press/v270/long25b.html.

Luigi Piccinelli, Christos Sakaridis, Mattia Segu, Yung-Hsu Yang, Siyuan Li, Wim Abbeloos, and Luc Van Gool. UniK3D: Universal camera monocular 3D estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1028–1039, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Piccinelli_UniK3D_Universal_Camera_Monocular_3D_Estimation_CVPR_2025_paper.html.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026a. URL https://qwen.ai/blog?id=qwen3.5.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026b. URL https://qwen.ai/blog?id=qwen3.8.

Santhosh K. Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X. Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-Matterport 3D dataset (HM3D): 1000 large-scale 3D environments for embodied AI, 2021. URL https://arxiv.org/abs/2109.08238.

Sonia Raychaudhuri, Saim Wani, Shivansh Patel, Unnat Jain, and Angel Chang. Language-aligned waypoint (LAW) supervision for vision-and-language navigation in continuous environments. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 4018–4028, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.328. URL https://aclanthology.org/2021.emnlp-main.328/.

Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 627–635. PMLR, 2011. URL https://proceedings.mlr.press/v15/ross11a.html.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A platform for embodied AI research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. URL https://openaccess.thecvf.com/content_ICCV_2019/html/Savva_Habitat_A_Platform_for_Embodied_AI_Research_ICCV_2019_paper.html.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025a. URL https://arxiv.org/abs/2503.11651.

Shuo Wang, Yongcai Wang, Wanting Li, Xudong Cai, Yucheng Wang, Maiyue Chen, Kaihui Wang, Zhizhong Su, Deying Li, and Zhaoxin Fan. Aux-Think: Exploring reasoning strategies for data-efficient vision-language navigation. In *Advances in Neural Information Processing Systems*, volume 38, pp. 34960–34984, 2025b. doi: 10.52202/085713-1040. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/2c90bf5bffe6497730ecade3a3a37458-Abstract-Conference.html.

Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, and Shuqiang Jiang. GridMM: Grid memory map for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15625–15636, October 2023a. URL https://openaccess.thecvf.com/content/ICCV2023/html/Wang_GridMM_Grid_Memory_Map_for_Vision-and-Language_Navigation_ICCV_2023_paper.html.

Zihan Wang, Xiangyang Li, Jiahao Yang, Yeqi Liu, Junjie Hu, Ming Jiang, and Shuqiang Jiang. Lookahead exploration with neural radiance representation for continuous vision-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13753–13762, June 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Wang_Lookahead_Exploration_with_Neural_Radiance_Representation_for_Continuous_Vision-Language_Navigation_CVPR_2024_paper.html.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 12009–12020, October 2023b. URL https://openaccess.thecvf.com/content/ICCV2023/html/Wang_Scaling_Data_Generation_in_Vision-and-Language_Navigation_ICCV_2023_paper.html.

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-and-language navigation. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/14da7aea05debb963b3d8d46449d51a0-Paper-Conference.pdf.

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. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, Vienna, Austria, June 2026b. URL https://arxiv.org/abs/2507.05240.

Yu Yan, Rongtao Xu, Jiazhao Zhang, Peiyang Li, Xiaodan Liang, and Jianqin Yin. InstruGen: Automatic instruction generation for vision-and-language navigation via large multimodal models. arXiv preprint arXiv:2411.11394, 2024. URL https://arxiv.org/abs/2411.11394.

Zhuoyuan Yu, Yuxing Long, Zihan Yang, Chengyan Zeng, Hongwei Fan, Jiyao Zhang, and Hao Dong. CorrectNav: Self-correction flywheel empowers vision-language-action navigation model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(22):18737–18745, 2026. doi: 10.1609/aaai.v40i22.38942. URL https://ojs.aaai.org/index.php/AAAI/article/view/38942.

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 *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/3812c3c803e986eca64000fe92467991-Abstract-Conference.html.

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 *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.079. URL https://roboticsproceedings.org/rss20/p079.html.

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. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025a. doi: 10.15607/RSS.2025.XXI.013. URL https://www.roboticsproceedings.org/rss21/p013.html.

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 *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13032–13056, Vienna, Austria, July 2025b. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.638. URL https://aclanthology.org/2025.acl-long.638/.

A THE USE OF LARGE LANGUAGE MODELS

We used GPT-5.6 and GPT-6 to edit manuscript language and layout. The authors review all edits and are responsible for the technical content, experimental results, and conclusions.

For dataset construction, Qwen3.8-27B (Qwen Team, 2026b) generates instructions from trajectory videos and compass images. We verify these against the observations, revise incorrect route or stopping descriptions, and discard samples that fail verification (Section 3.3).

B IMPLEMENTATION DETAILS

B.1 MODEL AND TRAINING SETTINGS

Table 6 lists model and training settings; Figure 9 gives the training and inference prompt.

Table 6: Default PanoVLN settings. Image dimensions are width $\times$ height.

Setting	Value
VLM / frozen geometry encoder	Qwen3.5-4B / PanoVGGT
Geometry features / fusion	Last aggregator layer; 2×2 grouping; fusion after the visual merger
Geometry normalization / projection	RMSNorm / MLP ($8192 \to 4096 \to 2560$, GELU)
Residual scale	$\alpha = 0.2$
Current / history ERP resize	960×480 / 448×224, before cropping
Visual-encoder ERP crop	20° from each pole
Geometry-encoder input	1036×518; full panorama
Visual history	Up to 10 earlier frames from the latest 100 observations
Optimizer / epochs	Fused AdamW / 1
Effective batch size	128 (8 GPUs $\times$ 4 examples $\times$ 4 accumulation steps)
Vision-encoder learning rate	2×10^{-6}
Language / merger / projector LR	2×10^{-5}
Learning-rate schedule / warmup	Cosine / 3% of training steps
Weight decay / gradient clipping	0.01 / max norm 10
Precision / random seed	bfloat16 / 42
Gradient checkpointing	Enabled
Prediction horizon / decoding	18 actions / greedy
CGE budget / minimum execution	$B = 1.2$ / $E_{\min} = 4$ actions
Action space	`forward`: 25 cm; `left`/`right`: 15°; `stop`

System

You are an autonomous navigation assistant. Your task is to follow the navigation instruction. Given the instruction, your recent observations, and your current observation, devise an action sequence using the four actions: left or right by 15 degrees, forward by 25 centimeters, or stop once the task is complete. Return exactly 18 action words in execution order, separated by spaces. If the task is complete before 18 actions, fill the remaining positions with stop.

User

Instruction: *[navigation instruction]*
History memory observations are panoramic views ordered from older to newer:
[history panorama 1] $\cdots$ *[history panorama m]*
Current observation (panoramic view):
[current panorama]
Devise the next action sequence.

Figure 9: Navigation prompt ($H = 18$). Italic fields are inputs; history is omitted when unavailable.

B.2 REAL-WORLD EXECUTION METRICS

Table 2 summarizes the time and planning costs of complete navigation trials under synchronous execution (Sec. 4.3). The robot requests a prediction, waits for the response, executes the selected actions, and then starts the next request. We compute the six metrics for each trial before averaging them across trials.

Navigation duration and speed. For trial i, let T_i be its elapsed navigation time in seconds and L_i its traveled distance in meters. The two navigation metrics are

$$\mathrm{Time}_i = T_i, \qquad \mathrm{Speed}_i = \frac{100 L_i}{T_i}\ \mathrm{cm/s}. \tag{5}$$

Time includes both action execution and waiting for policy responses. Speed measures progress along the executed trajectory per unit of elapsed time, with waiting included in the denominator. It therefore captures the combined effect of robot motion and interruptions during the trial.

Waiting time and motion interruptions. Let K_i be the number of policy requests and w_{ij} the elapsed time from issuing request j to receiving its response. Total policy-waiting time is $W_i = \sum_{j=1}^{K_i} w_{ij}$, giving

$$\mathrm{Wait}_i = 100 \frac{W_i}{T_i}\ \%. \tag{6}$$

Each waiting interval includes model inference and network communication. Wait measures the fraction of the trial consumed by these response intervals. Pauses counts continuous stationary intervals lasting longer than $1\,\mathrm{s}$ during the trial. Writing their durations as d_{ik},

$$\mathrm{Pauses}_i = \sum_k \mathbf{1}[d_{ik} > 1\,\mathrm{s}]. \tag{7}$$

A continuous interval is counted once. Wait describes the duration of policy-related delays, whereas Pauses describes the frequency of sustained motion interruptions. A response received within one second contributes to Wait even when it creates no pause longer than the threshold.

Policy calls and inference latency. Calls counts the requests issued during a trial. Let ℓ_{ij} be the model inference time for request j on the RTX 3090. We compute

$$\mathrm{Calls}_i = K_i, \qquad \mathrm{Latency}_i = \frac{1}{K_i} \sum_{j=1}^{K_i} \ell_{ij}\ \mathrm{s}. \tag{8}$$

Latency measures server-side model inference; the request–response interval used for Wait also includes communication. The measured mean network overhead is $272\,\mathrm{ms}$ per call. Calls and Latency describe the frequency and duration of planning, respectively, and together explain the policy-related waiting accumulated during navigation.

Aggregation. For each method, the table reports the arithmetic mean of each trial-level metric over the evaluated trials. Speed and Wait are computed separately for each trial and then averaged. Latency is first averaged over requests within a trial, then over trials. Pauses and Calls are integer counts for individual trials; their reported means can be fractional.

C MORE ABLATION STUDIES

These studies complement Sec. 4.4 by examining training-source composition and the VLM backbone. Both use the R2R-CE and RxR-CE Val-Unseen evaluation protocol in Sec. 4.1.

C.1 TRAINING DATA COMPOSITION

Table 7 adds training sources while holding Qwen3.5-4B, PanoVGGT, CGE, $H = 18$, and sample selection fixed. Starting from R2R-CE and RxR-CE, DAgger raises SR by 5.3 and 7.0 points and SPL by 4.2 and 6.7 points, respectively. This configuration corresponds to PanoVLN† in Table 1. Adding our constructed dataset further improves SR by 3.4/3.9 points and SPL by 2.7/2.0 points, yielding the full PanoVLN model. RxR-CE nDTW also increases from 71.4 to 73.3, indicating closer agreement with the instructed route alongside better completion.

The two added sources therefore provide complementary gains: our decision-centric data remains useful after DAgger has already strengthened the policy. This cumulative comparison assesses the full training recipe; Figure 8 complements it by comparing data construction strategies at matched trajectory counts.

Table 7: Training data composition on R2R-CE and RxR-CE Val-Unseen. All settings retain PanoVGGT, CGE, and the $H = 18$ prediction setting.

Training data				R2R-CE				RxR-CE			
R2R	RxR	DAgger	PanoVLN	NE↓	OS↑	SR↑	SPL↑	NE↓	SR↑	SPL↑	nDTW↑
✓	✓			3.91	73.6	68.6	63.7	4.02	67.1	57.2	71.5
✓	✓	✓		3.10	79.7	73.9	67.9	3.14	74.1	63.9	71.4
✓	✓	✓	✓	2.83	83.1	77.3	70.6	2.85	78.0	65.9	73.3

C.2 VISION-LANGUAGE BACKBONE

Table 8 evaluates the backbone choice (Bai et al., 2025a;b) with all four training sources, PanoVGGT, CGE, visual history, and $H = 18$ held fixed. Qwen3-VL-4B already achieves 76.2% SR on both benchmarks, with 70.4% and 64.9% SPL on R2R-CE and RxR-CE. Replacing it with Qwen3.5-4B improves SR by 1.1 and 1.8 points, respectively, and also improves the remaining reported metrics. These modest, consistent gains motivate our default backbone, while the strong Qwen3-VL-4B results show that PanoVLN's performance is not confined to a single VLM backbone.

Table 8: VLM backbone comparison on R2R-CE and RxR-CE Val-Unseen. All backbones use the full training-data configuration, PanoVGGT, CGE, and $H = 18$.

Backbone	R2R-CE				RxR-CE			
	NE↓	OS↑	SR↑	SPL↑	NE↓	SR↑	SPL↑	nDTW↑
Qwen2.5-VL-7B	2.96	82.2	75.9	68.6	2.88	76.3	65.1	72.7
Qwen3-VL-4B	2.91	81.5	76.2	70.4	2.96	75.7	64.9	71.6
Qwen3.5-4B	2.83	83.1	77.3	70.6	2.85	78.0	65.9	73.3

D TRAINING DATA ANALYSIS

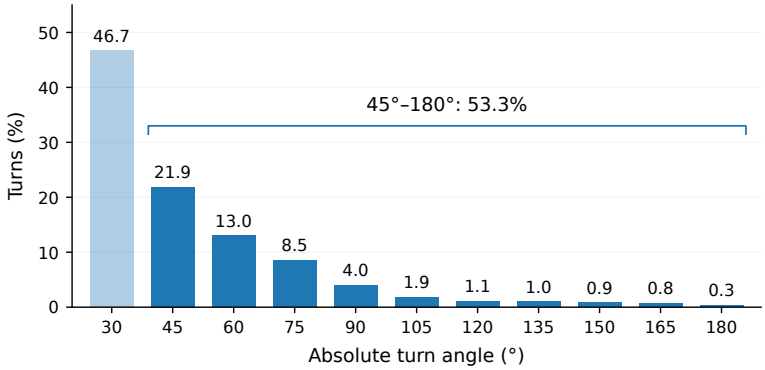

Figure 10: Reference-path turn distribution in R2R-CE and RxR-CE, normalized over 54,457 turns of at least 30°. The 45°–180° categories account for 53.3% of turns.

Reference-path turns. Figure 10 summarizes directional changes along reference paths in R2R-CE and RxR-CE. The pooled distribution uses 15° angle categories and is normalized over 54,457 turns of at least 30°. The 45°–180° categories account for 53.3% of these turns, showing that substantial direction changes are common in the reference routes.

For a forward 90° view aligned with the incoming path, an outgoing direction beyond 45° lies outside the current field of view. ERP observations retain these directions before rotation, providing visual context for the route beyond the turn.

E QUALITATIVE RESULTS

E.1 PAIRED PANORAMIC ATTENTION CASES

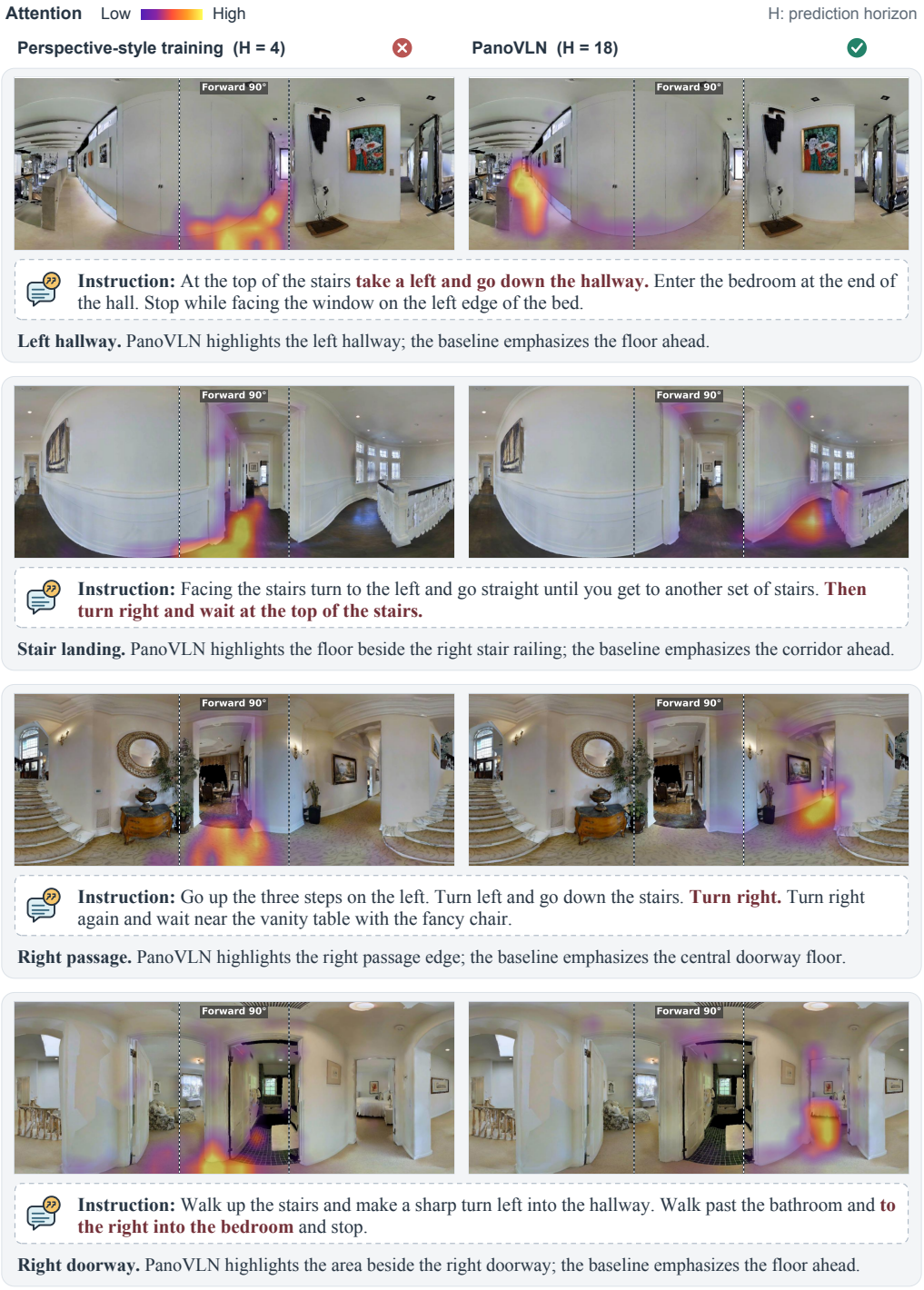

Figure 11: Additional paired attention cases, each using the same ERP and instruction. Blue bold text marks the relevant route clause; dashed lines bound the forward 90° sector. Each case note identifies the visible hotspot locations.

E.2 REAL-WORLD NAVIGATION CASES

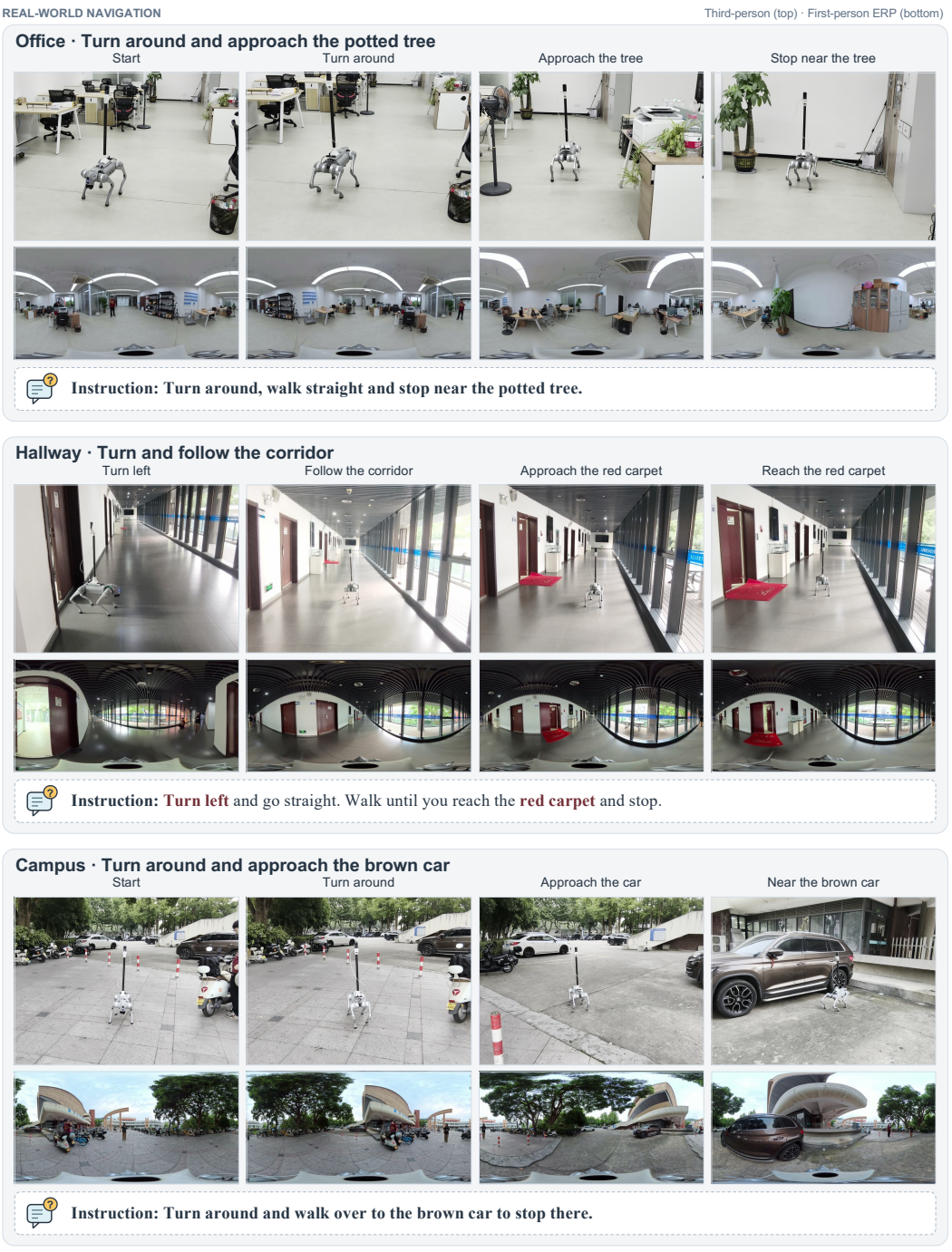

Figure 12: Real-world navigation cases in Hallway, Office, and Campus. Each row shows the instruction and successive observations from one trial, with timestamps and route annotations.

E.3 R2R-CE NAVIGATION CASES

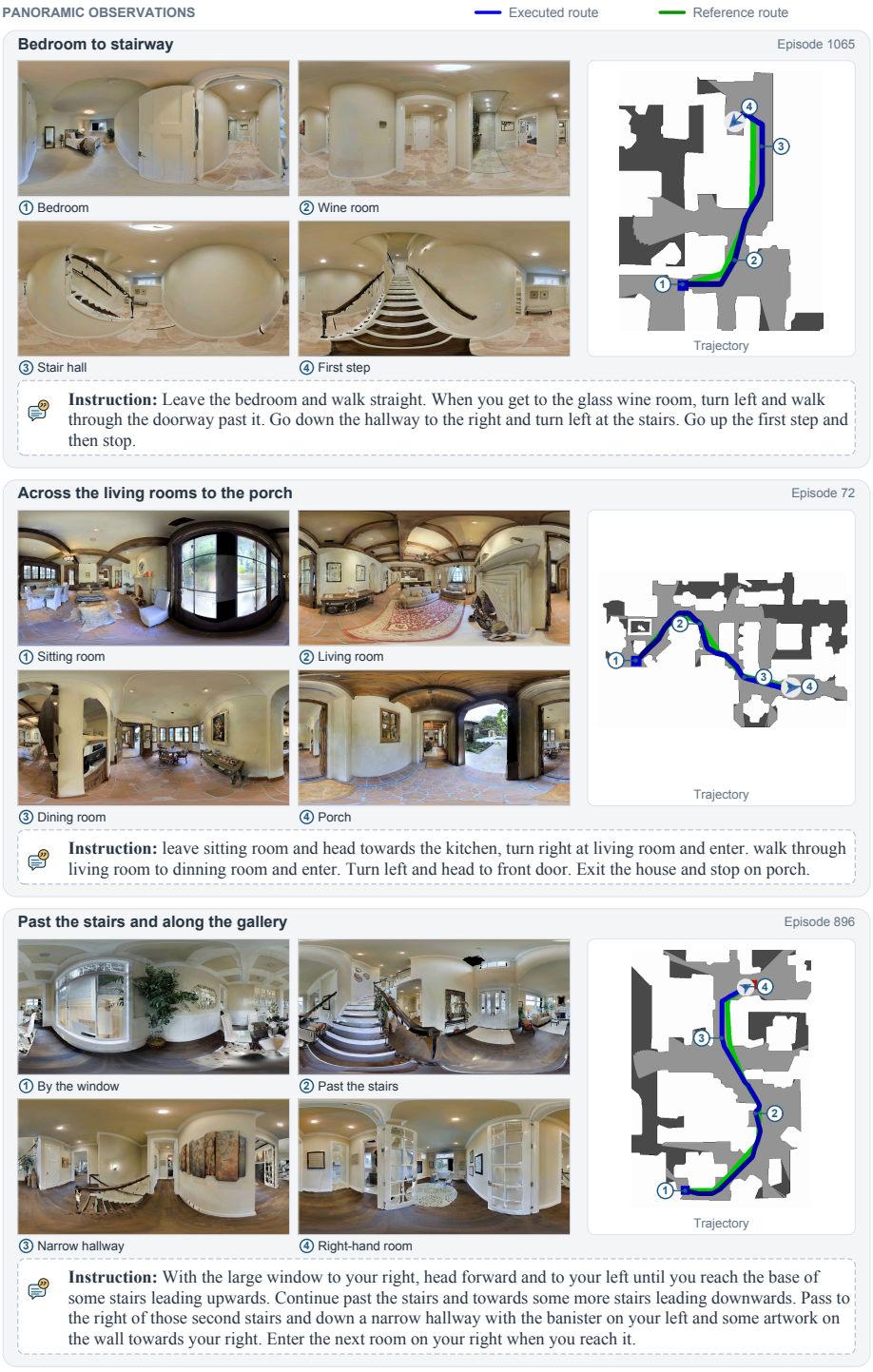

Figure 13: Navigation cases on R2R-CE Val-Unseen. Each case shows four panoramic observations in reading order and the full instruction. Numbered markers link each observation to its location on the trajectory; blue and green denote the executed and reference routes.

E.4 RxR-CE NAVIGATION CASES



Figure 14: Navigation cases on RxR-CE Val-Unseen. Four panoramic observations per case are linked to the trajectory by numbered markers. Blue and green denote the executed and reference routes. Instruction excerpts retain the original wording; ellipses mark omissions.