SMLA-NAV: END-TO-END LEARNING OF SPATIAL MEMORY, LOCALIZATION, AND ACTION FOR EMBODIED NAVIGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Service robots in long-term operational environments are often expected to navigate from arbitrary starting positions to destinations specified by text, images, or coordinates. However, jointly learning scene-specific spatial memory and navigation behavior from sparse goal-directed demonstrations remains challenging. We propose SMLA-Nav, an end-to-end spatial memory navigation framework that complements sparse expert trajectories with readily obtainable image–pose observations to acquire scene knowledge and adapt pretrained navigation skills. A consistent-prompt multi-stage training strategy bridges navigation pretraining and scene-specific post-training through a common spatial representation. This connects scene-specific localization and goal grounding with transferable point-goal navigation, composing these capabilities into a hierarchical spatial reasoning chain within a shared multimodal large language model (MLLM). At inference, the policy uses implicit spatial memory encoded in its parameters and deterministic geometric computation to navigate without querying external maps or localization databases. We establish SMLA-Bench, comprising SMLA-HM3D and SMLA-Gauss, on which SMLA-Nav achieves success rates of 86.9% and 85.7%, respectively. Comparisons using the same sparse scene trajectories further show improved text-goal navigation over direct MLLM post-training, while deployment on a quadruped robot demonstrates real-world feasibility.

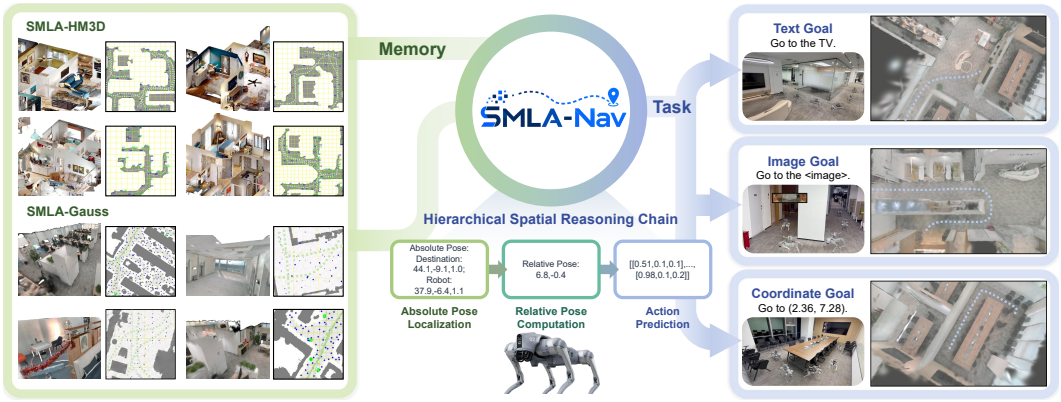


Figure 1: **SMLA-Nav** unifies absolute pose localization, tool-based relative pose computation, and short-horizon trajectory prediction around a shared MLLM. With implicit spatial memory, it navigates without explicit maps or external localization databases at inference, achieving 86.9% and 85.7% SR on SMLA-HM3D and SMLA-Gauss, respectively.

1 INTRODUCTION

Service robots in hotels, offices, and retail stores repeatedly navigate the same physical environments. Their users refer to particular places through names, images, or coordinates, and expect the robot to reach those places from arbitrary starting positions. Multimodal large language models

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.

(MLLMs) (Bai et al., 2025a;b; Li et al., 2024; Wang et al., 2025b) provide a natural foundation for these tasks through their ability to interpret diverse goal specifications and visual observations. At the same time, such goals often refer to specific physical locations in a particular scene, so successful navigation must combine transferable navigation skills with scene-specific spatial memory.

Existing systems address this need in different ways. Map-based methods (Zhu et al., 2025; Huang et al., 2026; Wang et al., 2025a) acquire explicit spatial memory through exploratory traversals to support goal-directed navigation, but errors may propagate across mapping, localization, and planning. Route-guided methods simplify decision making with externally provided route-level guidance, such as fine-grained language instructions (Zhang et al., 2024; 2025; Liu et al., 2026b) or waypoint sequences (Liu et al., 2025; Li et al., 2025), but such guidance is not always available for arbitrary start–goal pairs. Trajectory-based end-to-end policies (Liu et al., 2026a) further demonstrate that navigation behavior and implicit scene knowledge can be learned within a single MLLM from extensive expert demonstrations. However, collecting dense demonstrations is costly, while sparse trajectory supervision results in brittle performance. Hence, a central question remains: how can MLLMs learn implicit spatial memory for navigation under sparse trajectory supervision?

Our key insight is that scene-specific navigation can be learned from two complementary sources of supervision: dense image–pose observations for spatial grounding and sparse expert trajectories for navigation behavior. We consider a scene-adaptation setting in which image–pose pairs can be readily collected during pre-deployment exploratory traversals without requiring optimal actions, while only a few goal-directed expert trajectories are used. Through visual localization, these pairs support scene-level spatial memory acquisition and destination grounding, while expert trajectories enrich this memory with scene-layout and route information and adapt the pretrained navigation prior to the deployment scene.

We propose **SMLA-Nav**, a unified spatial reasoning framework with a shared MLLM backbone, as illustrated in Fig. 1. Our consistent-prompt multi-stage training strategy bridges navigation pretraining and scene-specific post-training by retaining a common spatial prompt structure while varying the input and supervised fields across training stages and data types. This integrates complementary image–pose and trajectory supervision, supporting spatial memory acquisition and goal grounding while helping retain pretrained navigation skills. Viewpoint augmentation and contrastive permutation further strengthen localization. At inference, these capabilities form a hierarchical spatial reasoning chain that connects localization with trajectory prediction through relative positions computed by a deterministic tool. The policy uses implicit spatial memory encoded in its parameters without querying explicit maps or external localization databases. For evaluation, we construct **SMLA-Bench**, comprising **SMLA-HM3D** and **SMLA-Gauss**, for navigation with text, image, and coordinate goals. SMLA-Nav achieves 86.9% and 85.7% SR on the two subsets, respectively, with corresponding SPL scores of 65.6% and 78.4%. Quadruped experiments further demonstrate real-world feasibility.

Our contributions are threefold. First, we introduce a scene-adaptation paradigm that uses dense image–pose pairs and sparse expert trajectories to acquire parametric spatial memory, ground goals, and adapt pretrained navigation skills. Second, we propose a consistent-prompt multi-stage training strategy that bridges navigation pretraining and scene-specific post-training through a common spatial structure, composing localization, tool-based relative-goal computation, and action prediction into a hierarchical spatial reasoning chain. Third, we establish SMLA-Bench for navigation with text, image, and coordinate goals and validate SMLA-Nav in simulation and on a quadruped robot.

2 Related Work

Explicit Memory Navigation. Many navigation methods store scene knowledge in explicit representations for use during decision making. Online approaches (Cao et al., 2025; Huang et al., 2026; Jiang et al., 2025; Peng et al., 2025; D. et al., 2026; Yang et al., 2025; Yin et al., 2024; 2025; Yu et al., 2025; 2023; Zhu et al., 2025; Zhao et al., 2026; Yang et al., 2026b) construct and update maps or structured memories while exploring unfamiliar environments to support subgoal selection and navigation. Other methods (Peng et al., 2026; Long et al., 2026; Ren et al., 2026; Wang et al., 2025a) build reusable scene representations before navigation, separating scene acquisition from execution as in our setting. However, their pipelines rely on consistent intermediate estimates across mapping, localization, goal retrieval, and planning, with errors potentially affecting route execution.

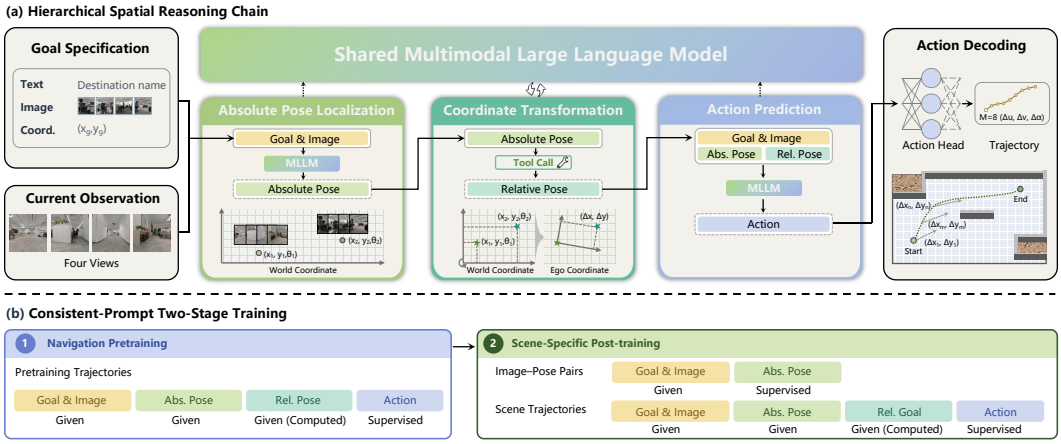


Figure 2: **Overview of SMLA-Nav.** (a) The hierarchical spatial reasoning chain localizes the goal and agent, computes their relative pose, and predicts navigation actions. (b) The consistent-prompt multi-stage strategy consists of navigation pretraining and scene-specific post-training.

Route-Guided Navigation. Route-guided navigation conditions action generation on visual observations and detailed language instructions (Zhang et al., 2024; 2025; Liu et al., 2026b) or waypoint sequences (Liu et al., 2025; Li et al., 2025). Although this setting differs from our destination-only task, end-to-end MLLM policies within this paradigm motivate our work.

Implicit Memory Navigation. Implicit memory methods encode spatial information in latent states or model parameters. Efficient-VLN (Zheng et al., 2025) uses progressive compression or learnable recursive memory to represent observation history during navigation. NavGSim (Liu et al., 2026a) demonstrates that MLLMs can acquire scene understanding from extensive expert trajectories, which are difficult to collect at scale in real-world environments. Under sparse trajectory supervision, jointly learning spatial memory and navigation control remains challenging. In contrast, SMLA-Nav integrates implicit spatial memory and a transferable point-goal policy within a shared MLLM backbone. A geometric interface connects visual localization to action prediction. Scene-specific post-training learns visual localization and goal grounding from complementary image–pose pairs and sparse trajectories while adapting the pretrained navigation policy using sparse trajectories.

3 Method

3.1 Task Definition

We study goal-directed navigation in previously observed scenes, with scene-specific post-training before deployment. Given a destination specified by text, an image, or world coordinates, the agent must reach it without collision. At inference, the policy uses no explicit scene map, external localization database, or online SLAM. At each step t, the agent receives four RGB views $\mathbf{V}_t = \{\mathbf{v}_t^j\}_{j=1}^4$ at 90° intervals and a goal $\mathbf{g}$. We use a Markovian policy conditioned only on $\mathbf{V}_t$ and $\mathbf{g}$, without prior frames. It predicts a short-horizon trajectory $\mathbf{T}_t = \{\mathbf{a}_{t,m}\}_{m=1}^M$, where M is the number of future steps and $\mathbf{a}_{t,m} = (\Delta u, \Delta v, \Delta\alpha) \in \mathbb{R}^3$ consists of an egocentric translation $(\Delta u, \Delta v)$ and a yaw rotation $\Delta\alpha$.

3.2 Overview

SMLA-Nav extends Qwen3-VL (Bai et al., 2025a) into a hierarchical spatial reasoning framework for spatial memory navigation, as shown in Fig. 2. Its policy is

$$\mathbf{T}_t = \pi_\Theta(\mathbf{V}_t, \mathbf{g}) \in \mathbb{R}^{M\times 3}, \tag{1}$$

where $\Theta = \{\psi, \phi\}$ comprises the parameters of the shared MLLM f_ψ and action head f_ϕ^{act}. The visual input and goal are embedded as $\mathbf{E}_t^{\text{V}}$ and $\mathbf{E}_{\text{L}}$, respectively. Using implicit spatial memory, the model localizes the agent and grounds text or image goals in world coordinates. Coordinate goals are supplied directly. A deterministic geometric adapter converts the absolute poses into an agent-centric relative pose. The shared MLLM then produces an action-token representation $\boldsymbol{\tau}_t$, which

the MLP action head decodes into $\mathbf{T}_t$. This connects goal grounding and scene-specific localization with a transferable point-goal policy. Sections 3.3 and 3.4 detail spatial memory and the reasoning chain, while Sec. 3.5 describes the consistent-prompt multi-stage training strategy.

3.3 Visual Localization-based Spatial Memory Mechanism

To achieve an implicit memory mechanism which enables flexible scene adaptation, SMLA-Nav internalizes spatial memory through visual localization (Kendall et al., 2015; Arandjelovic et al., 2016; Sarlin et al., 2019). We train SMLA-Nav with easily accessible image-location pairs $\mathcal{D}_{\text{mem}} = \{(\mathbf{V}_i, \mathbf{p}_i)\}_{i=1}^{N}$, where N is the number of samples and $\mathbf{p}_i = [x_i, y_i, \theta_i]^\top$ denotes the 2D location and heading in a shared scene coordinate frame. Through this supervision, the model learns to infer poses from novel viewpoints and retains reusable scene knowledge in its parameters. At inference, this knowledge can be reused across navigation episodes without querying external maps or localization databases.

Viewpoint Augmentation and Contrastive Permutation. Since localization quality directly determines the reliability of spatial memory, we further propose mechanisms to enhance localization accuracy under sparse training data. Four-view observations provide panoramic context and reduce spatial ambiguity. Heading diversity is further augmented by cyclically rotating the camera order. For $\mathbf{V}_i = \{\mathbf{v}_i^j\}_{j=1}^4$, the k-th rotated observation is defined as $\mathcal{R}_k(\mathbf{V}_i) = \{\mathbf{v}_i^{((j+k-1) \bmod 4)+1}\}_{j=1}^4$, where $k \in \{0, 1, 2, 3\}$. This treats each view as the front view in turn while keeping the position unchanged, and updates the heading annotation as $\theta_i^{(k)} = \text{wrap}(\theta_i - \frac{k\pi}{2})$, where $\text{wrap}(\cdot)$ normalizes the angle to $[-\pi, \pi)$. Based on the viewpoint-augmented observations, we pair corresponding elements of two within-scene permutations to construct goal–agent localization examples. Each example contains two observations and supervises both absolute poses, with every observation appearing once in each role. This preserves the underlying observation set while aligning localization training with the goal–agent pose format used in navigation.

3.4 Hierarchical Spatial Reasoning Chain

SMLA-Nav converts implicit spatial memory into executable actions through a hierarchical spatial reasoning chain. It makes spatial dependencies explicit through a structured process that first localizes the goal and the agent, then computes their relative spatial relationship, and finally determines how to act.

Specifically, SMLA-Nav first estimates the absolute poses of the goal and the agent in the world coordinate frame, denoted as $\hat{\mathbf{p}}^g = [\hat{x}^g, \hat{y}^g, \hat{\theta}^g]^\top$ and $\hat{\mathbf{p}}_t^r = [\hat{x}_t^r, \hat{y}_t^r, \hat{\theta}_t^r]^\top$, where the goal may be specified by text, image, or coordinate. For coordinate goals, the supplied goal position is used directly. The corresponding agent-centric relative position is denoted as $\Delta\hat{\mathbf{p}}_t = [\Delta\hat{x}_t, \Delta\hat{y}_t]^\top$. We compute it using a deterministic numerical tool, termed the Geometric Goal Adapter (GGA) and denoted by $\mathcal{G}$:

$$\Delta\hat{\mathbf{p}}_t = \mathcal{G}\big(\hat{\mathbf{p}}^g, \hat{\mathbf{p}}_t^r\big) = \begin{bmatrix} \sin\hat{\theta}_t^r & -\cos\hat{\theta}_t^r \\ -\cos\hat{\theta}_t^r & -\sin\hat{\theta}_t^r \end{bmatrix} \begin{bmatrix} \hat{x}^g - \hat{x}_t^r \\ \hat{y}^g - \hat{y}_t^r \end{bmatrix}. \tag{2}$$

This conversion uses the goal position and agent pose to obtain forward and rightward goal displacements under the coordinate conventions detailed in Appendix A.

The computed relative position is serialized into the relative-goal field and appended to the existing model context. Conditioned on this field, the preceding pose fields, and the original visual and goal inputs, the same MLLM continues inference to predict navigation actions. The complete reasoning process is summarized as

$$(\mathbf{V}_t, \mathbf{g}) \xrightarrow{f_\psi} (\hat{\mathbf{p}}^g, \hat{\mathbf{p}}_t^r) \xrightarrow{\mathcal{G}} \Delta\hat{\mathbf{p}}_t \xrightarrow{f_\psi} \boldsymbol{\tau}_t \xrightarrow{f_\phi^{\text{act}}} \mathbf{T}_t \in \mathbb{R}^{M\times 3}. \tag{3}$$

Here, f_ψ denotes the shared MLLM backbone, and f_ϕ^{act} maps its hidden representation $\boldsymbol{\tau}_t$ at the action token to an M-step motion sequence.

By explicitly modeling the intermediate spatial reasoning process, SMLA-Nav separates scene-specific spatial grounding from transferable point-goal navigation. Relative-position conditioning expresses the destination in a common agent-centric format across scenes and goal modalities,

whereas absolute pose localization depends on the layout and visual appearance of the deployment scene. Therefore, the point-goal navigation prior can be learned before scene adaptation and reused during post-training, while scene-specific supervision learns localization and goal grounding and adapts navigation behavior. The pretraining comparison in Appendix B.2 supports this design: absolute-plus-relative conditioning achieves higher SR and SPL than absolute-only conditioning, with performance comparable to relative-only conditioning.

3.5 Consistent-Prompt Multi-Stage Training Strategy

To learn the hierarchical reasoning chain, SMLA-Nav adopts a consistent-prompt multi-stage training strategy, as shown in Fig. 2. The prompt retains absolute-pose, relative-goal, and action-prediction fields while training progresses from transferable point-goal navigation to scene-specific localization and goal grounding. Relative poses use deterministic coordinate conversion as described in Sec. 3.4.

For K target absolute-pose tokens $\mathbf{y} = (y_1, \ldots, y_K)$, the localization cross-entropy and trajectory MSE are

$$\mathcal{L}_{\text{abs}} = -\frac{1}{K}\sum_{k=1}^{K} \log p_\psi(y_k \mid \mathbf{V}_t, \mathbf{g}, \mathbf{y}_{<k}), \quad \mathcal{L}_{\text{act}} = \frac{\|\mathbf{T}_t - \mathbf{T}_t^{\text{gt}}\|_F^2}{3M}, \tag{4}$$

where p_ψ is the shared MLLM's token distribution, $\mathbf{T}_t^{\text{gt}}$ is the reference trajectory, and $\|\cdot\|_F$ denotes the Frobenius norm.

Stage 1: Point-Goal Navigation Pretraining. Current visual observations, ground-truth absolute poses, and corresponding relative poses condition short-horizon action prediction. Training on diverse navigation data learns a transferable mapping from visual observations and agent-centric goal displacements to motion sequences.

Stage 2: Scene-Specific Post-Training. Scene adaptation uses image-location pairs $\mathcal{D}_{\text{mem}}$ and sparse goal-directed demonstrations $\mathcal{D}_{\text{nav}}$. Image–pose pairs supervise visual localization in scene coordinates and provide coverage beyond demonstrated routes. Trajectory-derived examples additionally associate annotated destination names and goal observations with their locations, while supervising navigation actions. Under teacher forcing, each trajectory sequence contains the goal specification, current observation, ground-truth absolute poses, computed relative pose, and action token. Only absolute-pose tokens incur cross-entropy, while relative poses remain as context without a prediction loss. The MLP decodes the action-token representation into a motion sequence supervised by MSE. Thus, image-location samples contribute $\mathcal{L}_{\text{abs}}$, while trajectory samples contribute both losses. The stage-wise objectives are

$$\mathcal{L}_{\text{pre}} = \mathcal{L}_{\text{act}}, \qquad \mathcal{L}_{\text{post}} = \mathcal{L}_{\text{abs}} + \lambda_{\text{act}}\mathcal{L}_{\text{act}}. \tag{5}$$

Losses are averaged over applicable samples. λ_{act} weights action supervision. The consistent prompt preserves the action interface conditioned on the relative pose, supporting scene-specific localization and goal grounding while helping maintain and adapt the pretrained navigation policy.

3.6 Data

Pretraining. We use approximately 4 million point-goal navigation examples from InteriorGS (Miao et al., 2025), VLN-CE trajectories based on RxR (Krantz et al., 2020; Ku et al., 2020), and Gaussian-based simulated environments to learn a transferable navigation prior.

Post-training and Benchmark. SMLA-Bench supports scene adaptation and evaluation. SMLA-HM3D contains 46 scenes from 30 annotated HM3D buildings (Ramakrishnan et al., 2021; Yadav et al., 2023) and 5,264 evaluation episodes. SMLA-Gauss contains 10 self-scanned indoor scenes reconstructed following NavGSim (Liu et al., 2026a) and 1,222 evaluation episodes. Scanned pretraining scenes use the same reconstruction pipeline but are disjoint from the benchmark scenes. Post-training uses only benchmark-provided posed observations and 1–3 sparse trajectories per destination. The observations contain four 224×224 views with a 90° field of view and are sampled on a $0.5\,\text{m} \times 0.5\,\text{m}$ grid. Since localization supervision requires only images and their associated poses, we collect these pairs directly through grid sampling in simulation, avoiding the need to implement an exploratory navigation policy. The trajectories cover representative routes rather than all start–goal paths. Figure 3 visualizes the data. Complete statistics and protocols are provided in Appendix A.

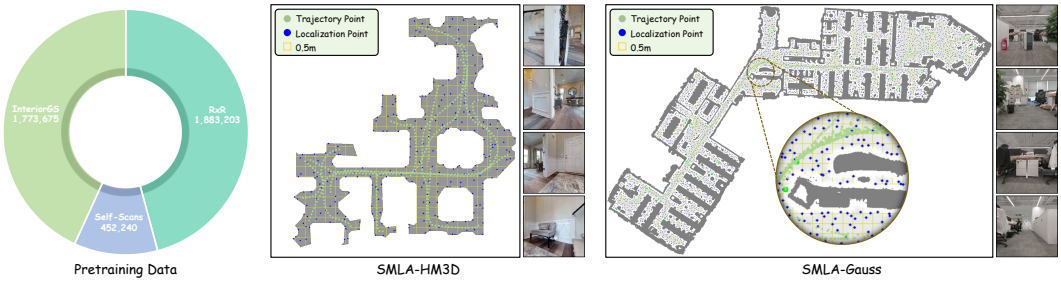


Figure 3: **Visualization of Pretraining Data and Benchmarks.** Left: pretraining data sources. Middle and right: representative SMLA-HM3D and SMLA-Gauss scenes with trajectory and localization points.

4 Experiments

4.1 Experiment Setup

Evaluation Protocol and Metrics. All experiments use SMLA-Bench (Sec. 3.6), with experiment-specific settings in Appendix B.1. We report Success Rate (SR) and Success weighted by Path Length (SPL) for navigation success and efficiency, alongside Collision Rate (CR), Stop Failure Rate (SFR), and Too-Long Rate (TLR) for failure analysis. Stop Failure at Alternative Places (SFAP) measures the proportion of stop failures occurring near predefined incorrect destinations. Unless otherwise stated, navigation results use episode-level micro-averaging, with rates and SPL reported as percentages. Localization is evaluated using Euclidean position error and absolute wrapped yaw error. Throughout the result tables, arrows indicate the preferred direction of each metric, while bold and underlined values mark the best and second-best primary results, respectively, within each comparison. Termination rules and complete metric definitions appear in Appendix A.5.

Implementation Details. SMLA-Nav is built upon Qwen3-VL-8B, with $M = 8$ action steps and $\lambda_{\text{act}} = 10$. Navigation pretraining takes approximately 8 hours on eight NVIDIA H200 GPUs, while joint post-training over all benchmark scenes takes approximately 12 hours on the same hardware. For real-world deployment, we perform site-specific post-training on real-robot data for approximately one hour and stream multi-view RGB observations to a remote server for waypoint prediction.

4.2 Quantitative Comparison

Navigation across Goal Modalities. To evaluate navigation across goal modalities, we compare SMLA-Nav with existing methods on SMLA-HM3D and SMLA-Gauss using text, image, and coordinate goals. Baselines are fine-tuned when their released interfaces support our trajectory supervision. Otherwise, their zero-shot settings are retained. As shown in Table 1, SMLA-Nav achieves the highest SR across all three goal modalities, reaching 84.3–89.7% on SMLA-HM3D and 79.7–90.4% on SMLA-Gauss, together with competitive SPL, demonstrating effective navigation to multimodal destinations. Zero-shot performance also depends on compatibility with our location-specific tasks. For AstraNav (Ren et al., 2025), our destination names refer to particular places whose locations cannot necessarily be inferred from everyday semantics alone, limiting its applicability without scene-specific grounding. Although fine-tuned, NavGSim (Liu et al., 2026a) and NavFoM (Zhang et al., 2025) achieve limited success, suggesting difficulty in acquiring spatial memory and goal-directed behavior from sparse trajectories alone. Differences in backbone capabilities and pretraining may also contribute to this gap. MM-Nav (Xu et al., 2025) and X-NavDP (Yang et al., 2026a) receive ground-truth relative goal positions at each step, bypassing absolute localization and facilitating efficient navigation. Further implementation details and analysis are provided in Appendix C.1.

Comparison with Preconstructed Explicit Memories. To compare implicit and preconstructed explicit memory-based navigation, we evaluate SMLA-Nav against modular systems using the same scene observations, poses, and sparse trajectories. Reference-map variants additionally use A^* on the evaluator occupancy map with retrieval-based localization (RAVEN), retrieval with feature matching and geometric verification (HLoc), or parametric pose prediction (SMLA-Nav). Ideal

Table 1: **Quantitative Comparison on SMLA-HM3D and SMLA-Gauss.** All values are percentages. † Ground-truth relative poses are supplied at each step. ‡ AstraNav's zero CR reflects its execution interface rather than demonstrated collision avoidance.

Task	Method	Setting	SMLA-HM3D SR↑	CR↓	SFR↓	TLR↓	SPL↑	SMLA-Gauss SR↑	CR↓	SFR↓	TLR↓	SPL↑
Text Goal	NavGSim (Liu et al., 2026a)	Fine-tuned	32.4	31.0	29.0	7.6	23.2	22.3	39.1	28.9	9.6	15.2
	NavFoM (Zhang et al., 2025)	Fine-tuned	40.2	24.4	28.0	7.4	27.2	28.9	35.5	26.9	8.6	21.2
	AstraNav (Ren et al., 2025)	Zero-shot	2.4	0.0‡	83.2	14.4	0.7	1.3	0.0‡	87.3	11.4	1.0
	SMLA-Nav	Fine-tuned	**89.7**	2.2	7.0	1.2	**68.0**	**90.4**	6.1	3.3	0.3	**86.0**
Image Goal	UniGoal (Yin et al., 2025)	Zero-shot	21.0	72.0	7.0	0.0	8.9	3.6	93.7	0.3	2.5	3.4
	T^2Nav (D. et al., 2026)	Zero-shot	30.1	60.8	9.1	0.0	11.8	3.6	93.7	0.3	2.5	3.5
	SMLA-Nav	Fine-tuned	**84.3**	3.0	9.9	2.8	**62.8**	**79.7**	14.7	5.6	0.0	**70.9**
Coordinate Goal	X-NavDP† (Yang et al., 2026a)	Zero-shot	31.8	68.2	0.0	0.0	26.4	53.1	46.9	0.0	0.0	44.4
	MM-Nav† (Xu et al., 2025)	Zero-shot	71.2	27.5	1.1	0.2	62.6	72.4	21.0	6.0	0.7	66.0
	SMLA-Nav	Fine-tuned	**86.8**	3.5	8.8	1.0	**65.9**	**86.9**	9.4	3.7	0.0	**78.3**

Table 2: **Comparison with Preconstructed Explicit Memories.** Reference-map configurations additionally receive the evaluator occupancy map.

Spatial Memory / Planner	Local Executor	SMLA-HM3D SR↑	CR↓	SFR↓	TLR↓	SPL↑	SMLA-Gauss SR↑	CR↓	SFR↓	TLR↓	SPL↑
Localization + Reference Map											
HLoc (Sarlin et al., 2019) + A^*	Ideal Polyline	74.2	25.7	0.2	0.0	74.2	80.4	7.1	12.4	0.0	79.8
HLoc (Sarlin et al., 2019) + A^*	SMLA-Nav (post-training)	77.9	19.7	2.4	0.0	**75.2**	82.6	15.0	2.5	0.0	**81.3**
RAVEN (Hu et al., 2026) + A^*	Ideal Polyline	58.9	24.6	7.0	9.6	55.8	71.8	11.5	11.2	5.5	68.1
RAVEN (Hu et al., 2026) + A^*	SMLA-Nav (post-training)	62.4	20.2	17.5	0.0	59.9	77.3	19.1	3.6	0.0	75.6
SMLA-Nav + A^*	Ideal Polyline	41.5	47.8	8.7	2.1	38.9	57.4	29.8	5.3	7.5	52.8
SMLA-Nav + A^*	SMLA-Nav (post-training)	85.4	5.9	7.3	1.4	65.6	83.5	13.8	2.7	0.0	76.7
Memory from Scene Data											
ULVN (Ren et al., 2026)	ViNT (Shah et al., 2023)	34.3	44.2	16.1	5.4	31.2	44.6	18.5	7.9	29.0	37.2
OpenNavMap (Jiao et al., 2026)	iPlanner (Yang et al., 2023)	50.8	28.9	20.3	0.0	43.8	55.7	32.6	11.7	0.0	47.8
MASt3R-Nav (Garg et al., 2026)	PixelReact	13.8	58.2	28.0	0.0	12.1	21.4	58.8	19.6	0.1	17.6
SMLA-Nav	Learned Trajectory	**86.9**	2.9	8.6	1.7	65.6	**85.7**	10.1	4.2	0.1	78.4

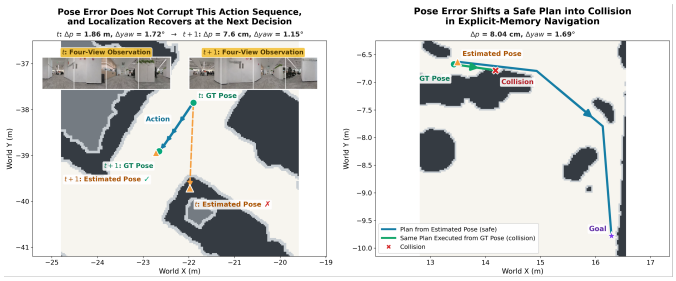


Figure 4: **Visualization of Navigation Results.** Left: SMLA-Nav avoids a collision and subsequently relocalizes. Right: executing motion planned from an estimated pose leads to collision from the actual pose.

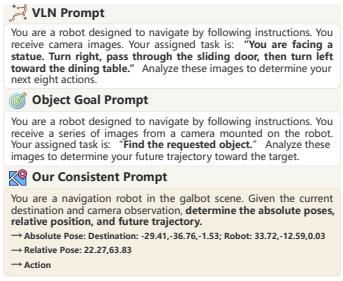


Figure 5: **Prompt Formats for Different Navigation Tasks.**

tracking follows the planned polyline, whereas our post-trained executor uses current images and a route waypoint about 5 m ahead. Both execute four steps (approximately 0.8 m) before relocalizing and replanning (Appendix C.2).

Table 2 shows that SMLA-Nav achieves 86.9% and 85.7% SR on SMLA-HM3D and SMLA-Gauss, versus 50.8% and 55.7% for OpenNavMap. Errors may arise at multiple stages of these modular pipelines and affect downstream navigation. Even with a reference map, motion planned from an estimated pose may cause a collision when executed from the actual pose, before the next replanning cycle. Figure 4 illustrates one such case: map-based tracking collides (right), whereas SMLA-Nav uses current visual observations to avoid a collision despite an inaccurate pose and subsequently relocalizes (left).

Table 3: **Comparison of Different Navigation Priors and Spatial Memory Formulations on SMLA-HM3D and SMLA-Gauss.** All variants are post-trained solely on the same sparse scene trajectories and evaluated with text goals.

Method	SMLA-HM3D						SMLA-Gauss					
	SR↑	CR↓	SFR↓	TLR↓	SPL↑	SFAP↓	SR↑	CR↓	SFR↓	TLR↓	SPL↑	SFAP↓
Qwen3-VL (Bai et al., 2025a)	59.8	12.3	24.5	3.4	40.5	77.1	44.2	30.5	12.7	12.7	21.8	72.0
Qwen3-VL-VLN	51.5	21.5	22.0	5.0	34.8	70.4	28.4	44.9	20.1	6.6	20.2	64.6
Qwen3-VL-OG	57.1	12.9	26.8	3.2	39.6	69.6	41.4	20.3	26.9	11.4	30.0	84.0
SMLA-Nav (traj. only)	**71.0**	2.9	21.8	4.3	**48.6**	**23.2**	**72.6**	12.7	10.7	4.1	**62.5**	**21.4**

Table 4: **Multi-Stage Navigation Evaluation on SMLA-HM3D and SMLA-Gauss.**

Stage	Scene Traj.	Absolute Pose GT	SMLA-HM3D					SMLA-Gauss				
			SR↑	CR↓	SFR↓	TLR↓	SPL↑	SR↑	CR↓	SFR↓	TLR↓	SPL↑
Pretraining	—	✓	79.2	19.8	1.0	0.0	70.7	78.1	16.7	4.8	0.4	71.2
Post-Training	×	✓	77.3	19.3	1.2	2.1	68.3	77.3	12.2	10.3	0.2	71.9
	×	×	62.9	27.5	8.2	1.4	43.2	64.7	15.5	16.5	3.3	57.4
	✓	✓	**92.0**	6.9	0.7	0.4	**75.4**	**90.1**	6.9	3.0	0.0	**85.4**
	✓	×	86.9	2.9	8.6	1.7	65.6	85.7	10.1	4.2	0.1	78.4

With each localizer and A^* fixed, replacing ideal tracking with our post-trained executor consistently improves SR on both benchmarks. Unlike ideal tracking, the learned executor predicts actions from current visual observations and a local point goal, allowing adaptation to scene states and reducing collision failures caused by route deviations. It further benefits from scene-specific navigation knowledge learned from sparse trajectories, highlighting the value of visually conditioned action prediction beyond direct motion reproduction. The full policy nevertheless achieves higher SR than these reference-map variants in Table 2 without querying external maps or localization databases. Together, these results support our destination-directed formulation, which combines transferable navigation skills with scene knowledge learned from sparse trajectories, beyond using the learned policy solely as a local waypoint executor.

Pretraining Priors and Spatial Memory Formulations. We evaluate how pretraining priors and spatial memory formulations affect scene adaptation. All methods are post-trained on the same sparse scene trajectories and evaluated with text goals. In this trajectory-only setting, SMLA-Nav constructs image-location pairs only from observations and poses along these trajectories. Figure 5 illustrates the different prompt formats, with VLN and OG denoting pretraining with VLN-format and object-goal-format data, respectively.

Table 3 shows that Qwen3-VL-VLN achieves lower SR and SPL than Qwen3-VL on both benchmarks, suggesting limited transfer of route-instruction-based action priors when post-training no longer provides step-by-step instructions. Qwen3-VL-OG improves SPL on SMLA-Gauss but lowers SR on both benchmarks, while its inconsistent SFAP changes suggest that matching the goal-description format alone does not ensure reliable goal-to-location binding. In contrast, SMLA-Nav increases SR from 59.8% to 71.0% on SMLA-HM3D and from 44.2% to 72.6% on SMLA-Gauss, with higher SPL on both. These results support the combined effectiveness of point-goal pretraining, consistent prompts, and pose-based spatial supervision for scene adaptation.

SMLA-Nav jointly learns localization and navigation from the same sparse trajectories, with pose supervision explicitly associating goals with scene locations. Compared with Qwen3-VL, SMLA-Nav reduces SFAP from 77.1% to 23.2% on SMLA-HM3D and from 72.0% to 21.4% on SMLA-Gauss. The lower SFAP further indicates reduced confusion among predefined destinations, suggesting more reliable goal-to-location binding.

4.3 Ablation Study

Visual Localization Ablation. We investigate how sampling resolution and localization components affect accuracy in the largest SMLA-Gauss scene, as shown in Fig. 6. Denser sampling generally reduces both position and yaw errors. For position estimation, adding four-view observations produces the largest incremental improvement at all four sampling resolutions, while rotation augmentation and contrastive permutation provide further gains. Relative to the single-view baseline,

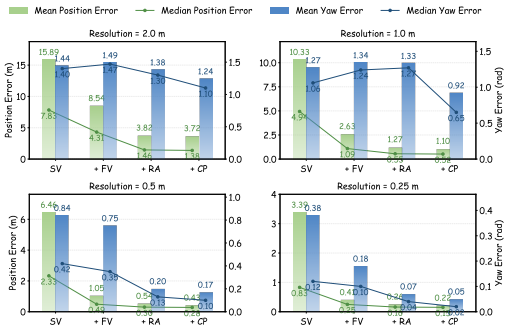


Figure 6: **Visual Localization Ablation Results.** SV, FV, RA, and CP correspond to single-view, four-view, rotation augmentation, and contrastive permutation, respectively.

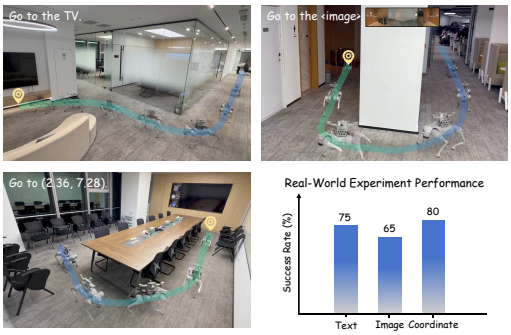


Figure 7: **Real-World Experiments Visualization.** Representative results are shown for the three navigation tasks.

the full configuration reduces mean yaw error from 1.44 to 1.24 rad at 2.0 m sampling and from 0.38 to 0.05 rad at 0.25 m sampling, showing a greater relative improvement under dense sampling.

Stage-wise Training Ablation. We evaluate whether the consistent-prompt multi-stage training strategy can preserve and compose the abilities learned across stages. As shown in Table 4, post-training without scene-specific trajectories, while replaying 80,000 motion steps from the pretraining data for action supervision, largely preserves navigation performance under ground-truth poses. Using sparse scene-specific trajectories further increases SR under ground-truth poses to 92.0% and 90.1% on SMLA-HM3D and SMLA-Gauss, suggesting that they provide additional scene-layout and route information beyond maintaining the navigation prior. Replacing ground-truth poses with predicted poses increases action MSE from 0.01066 to 0.01683 (Table 5), indicating that localization errors propagate to action prediction. Nevertheless, closed-loop navigation with predicted poses achieves SRs of 86.9% and 85.7% on the two subsets. Overall, the multi-stage strategy preserves the pretrained navigation ability and incorporates scene-specific spatial knowledge into the complete inference chain.

Table 5: **Error Propagation across Reasoning Stages.**

	Localization error		Action error
Abs. poses	XY (m)	Yaw (°)	Action MSE
GT	0	0	0.01066
Pred.	0.280	6.48	0.01683

Additional Analyses. We examine the effects of model parameter count and the use of LoRA (Appendix B.3), the roles of absolute-pose and relative-goal fields in point-goal pretraining (Appendix B.2), and robustness to view configuration, pose noise, grid spacing, start positions, and scene and route scale (Appendix D).

4.4 Real-World Experiments

We evaluate SMLA-Nav on a quadruped robot at a single adapted site, conducting 20 trials for each goal modality. SMLA-Nav achieves success rates of 75.0%, 65.0%, and 80.0% for text, image, and coordinate goals, respectively. All trials use the same physical start pose, with onboard obstacle avoidance enabled as a safety layer. These results demonstrate the real-world feasibility of SMLA-Nav under this deployment setting. Figure 7 presents representative trials, and Appendix E details scene adaptation, pose supervision, robot control, and observed failure cases.

5 Conclusion

We propose SMLA-Nav, an end-to-end framework that learns implicit spatial memory from complementary image–pose pairs and sparse trajectories within a shared MLLM. Its consistent-prompt training strategy connects scene grounding with pretrained navigation skills through a hierarchical spatial reasoning chain, enabling navigation without querying external maps or localization databases at inference. Experiments on SMLA-HM3D, SMLA-Gauss, and a quadruped robot demonstrate its effectiveness across text, image, and coordinate goals. Remaining limitations are discussed in Appendix F.

AI USE STATEMENT

Generative AI tools were used to assist with English translation and language editing, manuscript organization and drafting, LaTeX formatting, and figure and table presentation. They were not used to design or conduct experiments, or to generate experimental data. The authors reviewed and verified all AI-assisted content and take full responsibility for the final content of this work.

REFERENCES

Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5297–5307, 2016.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Ke qin 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. *ArXiv*, abs/2502.13923, 2025b. URL https://api.semanticscholar.org/CorpusID:276449796.

Yihan Cao, Jiazhao Zhang, Zhinan Yu, Shuzhen Liu, Zheng Qin, Qin Zou, Bo Du, and Kai Xu. Cognav: Cognitive process modeling for object goal navigation with llms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9550–9560, 2025.

Quang-Anh N. D., Duc Pham Minh, Minh-Anh Nguyen, Duy Tung Doan, and Tuan Dang. T2-nav: Algebraic-topologyaware temporal graph memory and loop detection for zero-shot visual navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2026.

Vansh Garg, Rohit Jayanti, Krish Pandya, Sarthak Chittawar, Siddharth Tourani, Muhammad Haris Khan, Sourav Garg, and Madhava Krishna. MASt3R-Nav: WayPixel navigation in relative 3d maps. *arXiv preprint arXiv:2605.24111*, 2026.

Yixun Hu, Zhicheng Zheng, Lihan Zha, Chunwei Xing, Rajdeep Singh, Omar Hossain, Antonio Loquercio, and Dhruv Shah. Raven: Long-horizon reasoning & navigation with a visuo-spatio-temporal memory. *arXiv preprint arXiv:2606.25206*, 2026.

Xun Huang, Shijia Zhao, Yunxiang Wang, Xin Lu, Wanfa Zhang, Rongsheng Qu, Weixin Li, Yunhong Wang, and Chenglu Wen. Msgnav: Unleashing the power of multi-modal 3d scene graph for zero-shot embodied navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 37154–37163, 2026.

Jiajun Jiang, Yiming Zhu, Zirui Wu, and Jie Song. Dualmap: Online open-vocabulary semantic mapping for natural language navigation in dynamic changing scenes. *IEEE Robotics and Automation Letters*, 2025.

Jianhao Jiao, Changkun Liu, Jingwen Yu, Boyi Liu, Qianyi Zhang, Yue Wang, and Dimitrios Kanoulas. Opennavmap: Multi-session appearance-based topometric mapping for scalable visual navigation. *IEEE Transactions on Robotics*, 2026.

Alex Kendall, Matthew Koichi Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2938–2946, 2015. URL https://api.semanticscholar.org/CorpusID:12888763.

Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision*, pp. 104–120. Springer, 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 *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 4392–4412, 2020.

Anqi Li, Zhiyong Wang, Jiazhao Zhang, Minghan Li, Yunpeng Qi, Zhibo Chen, Zhizheng Zhang, and He Wang. Urbanvla: A vision-language-action model for urban micromobility. *arXiv preprint arXiv:2510.23576*, 2025.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

Jiahang Liu, Yuanxing Duan, Jiazhao Zhang, Minghan Li, Shaoan Wang, Zhizheng Zhang, and He Wang. Navgsim: High-fidelity gaussian splatting simulator for large-scale navigation. *arXiv preprint arXiv:2603.15186*, 2026a.

Jiahang Liu, Tianyu Xu, Jiawei Chen, Lu Yue, Jiazhao Zhang, Zhiyong Wang, Minghan Li, Qisheng Zhao, Anqi Li, Qi Su, et al. Span-nav: Generalized spatial awareness for versatile vision-language navigation. *arXiv preprint arXiv:2603.09163*, 2026b.

Xinhao Liu, Jintong Li, Yicheng Jiang, Niranjan Sujay, Zhicheng Yang, Juexiao Zhang, John Abanes, Jing Zhang, and Chen Feng. Citywalker: Learning embodied urban navigation from web-scale videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 6875–6885, 2025.

Junlin Long, Zeyu Zhang, Xu Deng, Yiran Wang, Yue Yang, Luke Borgnolo, Maxwell Twelftree, and Yang Zhao. Platonicnav: Unveiling semantic correspondence in navigation with platonic topological maps. *arXiv preprint arXiv:2606.01788*, 2026.

Bingchen Miao, Rong Wei, Zhiqi Ge, Xiaoquan Sun, Shiqi Gao, Jingzhe Zhu, Renhan Wang, Siliang Tang, Jun Xiao, Rui Tang, and Juncheng Li. Towards physically executable 3d gaussian for embodied navigation. *ArXiv*, abs/2510.21307, 2025. URL `https://api.semanticscholar.org/CorpusID:282384265`.

Cheng Peng, Zhenzhe Zhang, Xiaobao Wei, Yanhao Zhang, Heng Wang, Pengwei Wang, Zhongyuan Wang, Cheng Chi, Shanghang Zhang, and Jing Liu. Pigeon: Vlm-driven object navigation via points of interest selection. *arXiv preprint arXiv:2511.13207*, 2025.

Daojie Peng, Bingtao Wang, and Jun Ma. Sstg-nav: Metric-grounded spatial-semantic topological graphs for reusable object navigation. *arXiv preprint arXiv:2608.00527*, 2026.

Santhosh K Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alex Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. *arXiv preprint arXiv:2109.08238*, 2021.

Botao Ren, Junjun Hu, Xinda Xue, Minghua Luo, Jintao Chen, Haochen Bai, Liangliang You, and Mu Xu. Astranav-memory: Contexts compression for long memory. *arXiv preprint arXiv:2512.21627*, 2025.

Hao Ren, Junzhe Zhu, Yihan Li, Zetong Bi, Le Zheng, Zhi Li, Yiqing Yuan, Zhaoliang Wan, Dizhe Zhang, Lu Qi, et al. Unordered landmark visual navigation. *arXiv preprint arXiv:2608.06833*, 2026.

Paul-Edouard Sarlin, Cesar Cadena, Roland Siegwart, and Marcin Dymczyk. From coarse to fine: Robust hierarchical localization at large scale. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 12708–12717. IEEE, 2019.

Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. ViNT: A foundation model for visual navigation. *arXiv preprint arXiv:2306.14846*, 2023.

Bo Wang, Jiehong Lin, Chenzhi Liu, Xinting Hu, Yifei Yu, Tianjia Liu, Zhongrui Wang, and Xiaojuan Qi. Mg-nav: Dual-scale visual navigation via sparse spatial memory. *arXiv preprint arXiv:2511.22609*, 2025a.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025b.

Tianyu Xu, Jiawei Chen, Jiazhao Zhang, Wenyao Zhang, Zekun Qi, Minghan Li, Zhizheng Zhang, and He Wang. Mm-nav: Multi-view vla model for robust visual navigation via multi-expert learning. *ArXiv*, abs/2510.03142, 2025. URL https://api.semanticscholar.org/CorpusID:281830207.

Karmesh Yadav, Ram Ramrakhya, Santhosh Kumar Ramakrishnan, Theo Gervet, John Turner, Aaron Gokaslan, Noah Maestre, Angel Xuan Chang, Dhruv Batra, Manolis Savva, et al. Habitat-matterport 3d semantics dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4927–4936, 2023.

Fan Yang, Chen Wang, Cesar Cadena, and Marco Hutter. iPlanner: Imperative path planning. *arXiv preprint arXiv:2302.11434*, 2023.

Tianyu Yang, Yiming Zeng, Wenzhe Cai, Yuqiang Yang, Jiaqi Peng, Hui Cheng, Jiangmiao Pang, and Tai Wang. X-NavDP: Generalizing navigation diffusion policy to novel behavior and embodiments with group q-score reweighted matching. *arXiv preprint arXiv:2607.28560*, 2026a.

Yuncong Yang, Han Yang, Jiachen Zhou, Peihao Chen, Hongxin Zhang, Yilun Du, and Chuang Gan. 3d-mem: 3d scene memory for embodied exploration and reasoning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17294–17303. IEEE, 2025.

Zebin Yang, Sunjian Zheng, Tong Xie, Tianshi Xu, Bo Yu, Fan Wang, Jie Tang, Shaoshan Liu, and Meng Li. Efficientnav: Towards on-device object-goal navigation with navigation map caching and retrieval. *Advances in Neural Information Processing Systems*, 38:4286–4312, 2026b.

Hang Yin, Xiuwei Xu, Zhenyu Wu, Jie Zhou, and Jiwen Lu. Sg-nav: Online 3d scene graph prompting for llm-based zero-shot object navigation. *ArXiv*, abs/2410.08189, 2024. URL https://api.semanticscholar.org/CorpusID:273233851.

Hang Yin, Xiuwei Xu, Linqing Zhao, Ziwei Wang, Jie Zhou, and Jiwen Lu. Unigoal: Towards universal zero-shot goal-oriented navigation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19057–19066. IEEE, 2025.

Bangguo Yu, Hamidreza Kasaei, and Ming Cao. L3mvn: Leveraging large language models for visual target navigation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3554–3560. IEEE, 2023.

Bangguo Yu, Qihao Yuan, Kailai Li, Hamidreza Kasaei, and Ming Cao. Co-navgpt: Multi-robot cooperative visual semantic navigation using vision language models. *IEEE Robotics and Automation Letters*, 11(2):2122–2129, 2025.

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*, 2024.

Jiazhao Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, et al. Embodied navigation foundation model. *arXiv preprint arXiv:2509.12129*, 2025.

He Zhao, Yijun Yang, Zichuan Lin, Deheng Ye, and Chunyan Miao. Hiro-nav: Hybrid reasoning enables efficient embodied navigation. *arXiv preprint arXiv:2604.08232*, 2026.

Duo Zheng, Shijia Huang, Yanyang Li, and Liwei Wang. Efficient-vln: A training-efficient vision-language navigation model. *arXiv preprint arXiv:2512.10310*, 2025.

Ziyu Zhu, Xilin Wang, Yixuan Li, Zhuofan Zhang, Xiaojian Ma, Yixin Chen, Baoxiong Jia, Wei Liang, Qian Yu, Zhidong Deng, et al. Move to understand a 3d scene: Bridging visual grounding and exploration for efficient and versatile embodied navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8120–8132, 2025.

APPENDIX

This appendix details the benchmark and evaluation protocol, the settings and baseline adaptations underlying the main-paper comparisons, and additional analyses of training, robustness, and real-world deployment.

A SMLA-BENCH: DATASET CONSTRUCTION AND EVALUATION PROTOCOL

A.1 BENCHMARK COMPOSITION AND GOAL INTERFACES

Benchmark Composition. SMLA-Bench consists of two complementary subsets, SMLA-HM3D and SMLA-Gauss. Both subsets follow the same location-specific navigation tasks, scene-specific post-training protocol, and evaluation criteria, while differing in their scene sources and simulation construction.

SMLA-HM3D contains 46 scenes derived from 30 HM3D buildings with semantic annotations (Ramakrishnan et al., 2021; Yadav et al., 2023). SMLA-Gauss contains ten indoor scenes collected and scanned by the authors, including offices and industrial environments. Following the environment-construction pipeline of NavGSim (Liu et al., 2026a), we reconstruct these scans as navigable Gaussian-based simulation environments. The underlying scenes are collected by the authors and are not taken from NavGSim.

Location-Specific Navigation Tasks. All three SMLA-Bench goal modalities identify a particular physical destination rather than an arbitrary instance of a semantic category.

For text-goal navigation, the input is a scene-specific destination name introduced through the post-training data. Multiple places may have similar semantic content, but only the named destination is considered correct.

For image-goal navigation, the goal is represented by a four-view RGB observation captured at the destination.

For coordinate-goal navigation, the destination is specified by its world-frame position (x, y).

These tasks evaluate whether a navigation system can associate linguistic, visual, and geometric goal descriptions with specific places and convert the resulting grounding into executable navigation behavior.

For text and image goals, goal grounding produces the destination pose before agent localization and relative-goal computation. For coordinate goals, the supplied world position is inserted directly into the goal-position field, bypassing destination inference. All three modalities consequently share the same action interface conditioned on the relative pose.

Coordinate and Action Conventions. The benchmark uses a 2D world coordinate frame in which the x-axis points rightward and the y-axis points upward. Robot yaw is zero when the robot faces the negative y direction, and positive world-frame yaw denotes a left turn.

The relative pose $(\Delta x, \Delta y)$ expresses forward and rightward displacement to the destination. Motion targets use the same egocentric axis convention, with separate symbols: Δu denotes forward displacement, Δv denotes rightward displacement, and $\Delta\alpha > 0$ denotes a right turn. These action increments describe predicted motion rather than the displacement to the final goal. A method may internally use individual actions, multi-step motion targets, an explicit planner, or another decision representation, provided that its output is executed under the same environment dynamics and evaluation protocol.

A.2 SCENE-SPECIFIC POST-TRAINING DATA

Post-Training Data Construction. The post-training data contain two complementary forms of scene-specific supervision: posed visual observations for localization and sparse goal-directed trajectories for navigation. These two data sources serve different purposes. Localization observations

provide broad visual coverage and supervise where the robot is in the scene, whereas navigation trajectories provide action supervision along a limited set of traversable routes.

The complete post-training sets contain 14,027 localization records and 788 navigation trajectories in SMLA-HM3D, and 6,070 localization records and 188 navigation trajectories in SMLA-Gauss.

Each observed trajectory point also supplies an image–pose pair for the localization objective. Thus, the same trajectory contributes two training streams: observation-to-pose supervision and goal-conditioned action supervision. In action-supervised examples, absolute poses are supplied and the relative pose is computed deterministically, preserving the point-goal interface used during pre-training.

Broad Visual Coverage for Scene Grounding. The localization records approximate visual observations that can be collected during non-goal-directed traversal of a deployment environment. Unlike navigation demonstrations, these observations do not specify how the robot should move toward a destination. Instead, they supervise observation-to-pose grounding, allowing the model to associate visual appearances with locations in the scene coordinate frame and estimate its current pose.

Casual traversal in a physical environment produces irregular spatial coverage and is difficult to reproduce consistently across simulated scenes. We therefore construct a controlled approximation by partitioning each navigable region into $0.5\,\mathrm{m} \times 0.5\,\mathrm{m}$ cells and sampling one valid position from each cell intersecting the navigable region. At every sampled position, we collect four RGB views of 224×224 pixels, each with a 90° field of view.

This construction provides broad visual exposure without treating the observations as goal-directed navigation demonstrations. In particular, a location may be covered by the localization data even when no post-training trajectory connects a particular start position to that location. Localization coverage and demonstration-route coverage therefore represent two distinct properties of the same observed scene.

Sparse Navigation Trajectories. We construct the navigation portion of the post-training set around scene-specific landmarks, each serving as a destination. For each landmark, we sample up to two valid starts in SMLA-HM3D and up to three in SMLA-Gauss. Fewer are retained when sufficient reachable starts are unavailable. Larger scenes receive more landmarks rather than more starts per landmark. The resulting 788 HM3D and 188 Gauss trajectories cover representative connections among landmarks and principal passages without enumerating all start–goal combinations.

Training routes are generated with a collision-aware procedure that favors clearance from obstacles and the interior of traversable regions. These routes provide scene-specific examples of traversable connections while maintaining the point-goal action interface learned during pretraining. They need not coincide with the shortest paths used for evaluation. Each trajectory is stored as a temporally ordered sequence. A navigation record associates the current four-view observation with the subsequent eight-step motion target, with increments represented as $(\Delta u, \Delta v, \Delta\alpha)$.

Construction of Text and Image Goals. Each collected trajectory is instantiated with two goal specifications while retaining the same motion sequence. The text-goal version uses the predefined name of its endpoint landmark, and the image-goal version uses an observation of that endpoint. The resulting pose supervision teaches name-to-location and image-to-location associations, respectively. The current text goals are fixed landmark names. Compositional spatial descriptions are left for future work. Coordinate goals supply the endpoint coordinates directly and therefore require no learned destination binding.

A.3 Evaluation Episodes and Information Access

Evaluation Episode Construction. SMLA-Bench evaluates text-goal, image-goal, and coordinate-goal navigation. We keep the numbers of evaluated destinations approximately balanced across the three goal modalities. The number of destinations within an individual scene is not fixed and varies with the available scene-specific targets.

Table 6: **Composition and Scale Distribution of SMLA-Bench.** Evaluation-path bins count unique evaluation episodes and use the reference geodesic shortest-path length from the evaluation start to the intended goal. They do not describe post-training trajectories or policy-executed paths. Scene-area bins count scenes and use navigable area. Percentages are normalized separately within each subset.

Group	Quantity or Interval	SMLA-HM3D	SMLA-Gauss
Composition	Number of scenes	46	10
	Evaluation episodes	5,264	1,222
	Post-training localization records	14,027	6,070
	Post-training trajectories	788	188
Evaluation path length	Short: $0 < \ell^{\star}_{\mathrm{eval}} \leq 10\,\mathrm{m}$	3,966 (75.34%)	555 (45.42%)
	Medium: $10 < \ell^{\star}_{\mathrm{eval}} \leq 30\,\mathrm{m}$	1,298 (24.66%)	268 (21.93%)
	Long: $30 < \ell^{\star}_{\mathrm{eval}} \leq 100\,\mathrm{m}$	0 (0.00%)	399 (32.65%)
Scene area	Small: $0 \leq A_{\mathrm{scene}} \leq 50\,\mathrm{m}^2$	26 (56.52%)	7 (70.00%)
	Medium: $50 < A_{\mathrm{scene}} \leq 300\,\mathrm{m}^2$	20 (43.48%)	2 (20.00%)
	Large: $300 < A_{\mathrm{scene}} \leq 800\,\mathrm{m}^2$	0 (0.00%)	1 (10.00%)

For each evaluated destination, we randomly sample multiple valid start positions to construct evaluation episodes. SMLA-HM3D uses approximately seven evaluation starts per destination, whereas SMLA-Gauss uses approximately ten. Each start–destination pair defines one evaluation episode. Consequently, the reported evaluation-episode counts refer to start–destination navigation queries rather than to the number of distinct destinations.

Different methods are evaluated using the same episode manifests and the same start–destination queries. Every unique evaluation episode receives equal weight when computing the reported navigation metrics.

Dataset Scale. Table 6 summarizes the composition of the two subsets and their distributions over evaluation-path length and scene size.

Here, $\ell^{\star}_{\mathrm{eval}}$ denotes the reference geodesic shortest-path length from an evaluation start position to its intended goal. It is used to characterize evaluation episodes and to compute SPL. It differs from both the length of a post-training trajectory and the path length executed by a policy during evaluation.

The symbol A_{scene} denotes navigable scene area. For SMLA-HM3D, we compute it separately for each floor from the planar projections of its navmesh triangles.

Separation of Post-Training and Evaluation Data. Post-training and evaluation use separate, deduplicated manifests. No reference action sequence belonging to an evaluation episode is used for post-training or hyperparameter selection. Evaluation outcomes are likewise excluded from model selection.

Because SMLA-Bench evaluates navigation within an observed deployment scene, scene identities and destination identities are intentionally shared between post-training and evaluation. Individual observed locations or route segments may also overlap. Each evaluation episode is treated as a separate navigation query, and its reference action sequence is withheld from post-training.

The primary protocol does not require every evaluation start pose to lie outside the spatial support of the localization observations, nor does it require every route segment to be absent from all post-training trajectories. The resulting evaluation measures navigation under a declared amount of scene exposure and should not be interpreted as zero-shot navigation in an entirely unseen environment.

Scene-Specific Information Constraints. SMLA-Bench does not prescribe how a method represents the observed scene. A method may use an explicit metric map, a topological graph, an image-retrieval memory, or implicit spatial memory encoded in model weights. For the primary matched-information comparison, however, every scene-specific representation must be constructed from

the same benchmark-provided observations, poses, destination bindings, and navigation demonstrations.

Additional scans, oracle scene layouts, reference occupancy maps, depth observations, evaluation poses, or scene-specific demonstrations are treated as additional resources and must be disclosed separately. Results obtained with these resources may be presented as reference comparisons or component analyses, but they are not treated as primary comparisons under matched scene-information access.

A.4 Map-Free Closed-Loop Inference

At each decision step, SMLA-Nav receives the current visual observation and goal specification without previous observations or actions. It first grounds a text or image goal, or inserts the supplied coordinate goal, then localizes the agent. The geometric goal adapter computes the relative goal, which is appended to the model context before action prediction. Implicit spatial memory is encoded in the navigation model's weights through scene-specific post-training. Here, map-free inference means that the policy does not query external scene representations, including maps, localization databases, or stored scene observations and trajectories. The geometric adapter only converts coordinates. It does not search for a route.

From Short-Horizon Actions to Long-Horizon Navigation. The action head predicts an eight-step motion sequence, but an evaluation route may require many such predictions. After local execution, the model uses the updated observation to relocalize and predict the next motion sequence toward the same destination. Sparse scene trajectories provide examples of local motion and scene connectivity, and repeated closed-loop decisions compose these short-horizon predictions into longer routes. This procedure does not require a complete route to be generated in one forward pass. Its dependence on route length is evaluated in Appendix D.5.

The current implementation recomputes text- and image-goal poses at each decision step. This preserves a common prompt structure, but repeats goal grounding for a fixed destination. The resulting computational redundancy is discussed in Appendix F.

A.5 Termination and Evaluation Metrics

Episode Termination. An episode receives exactly one of four terminal labels: success, collision, stop failure, or too-long failure. SMLA-Nav does not predict a separate stop token. Instead, stopping is inferred from the predicted motion between the current pose and the first pose of the predicted trajectory. Let $\mathbf{p}_{t,0}, \alpha_{t,0}$ denote the current planar position and heading, and let $\mathbf{p}_{t,1}, \alpha_{t,1}$ denote the first predicted pose. We compute

$$d_t^{\text{move}} = \|\mathbf{p}_{t,1} - \mathbf{p}_{t,0}\|_2, \qquad d_t^{\text{angle}} = |\text{wrap}(\alpha_{t,1} - \alpha_{t,0})|, \tag{6}$$

and trigger stopping when

$$d_t^{\text{move}} < 0.005\,\text{m} \quad \wedge \quad d_t^{\text{angle}} < 0.03\,\text{rad}. \tag{7}$$

A triggered stop is labeled as success when the terminal robot position lies within the dataset-specific success radius of the intended destination: $1.0\,\text{m}$ for SMLA-HM3D and $1.5\,\text{m}$ for SMLA-Gauss. Otherwise, it is labeled as a stop failure. A collision terminates the episode when the agent attempts to enter a non-traversable region. An episode whose accumulated executed path length reaches $150\,\text{m}$ without receiving an earlier terminal label is labeled as a too-long failure.

Reference Paths and SPL. SPL on SMLA-HM3D uses Habitat's geodesic shortest-path length as the evaluation reference. In contrast, the collision-aware training routes favor obstacle clearance and often follow the interior of traversable regions, so they need not coincide with the length-optimal reference paths. This mismatch may encourage longer executed routes and partly explain the lower SPL on SMLA-HM3D than on SMLA-Gauss. The collected routes and their scene coverage are visualized in Fig. 3.

Navigation Metrics and Aggregation. Let N be the number of unique evaluation episodes, let $S_i \in \{0, 1\}$ indicate whether episode i succeeds, let ℓ_i denote its policy-executed path length, and let $\ell_i^\star$ denote its reference geodesic shortest-path length. We compute

$$\mathrm{SR} = 100\frac{N_{\mathrm{SR}}}{N}\%, \tag{8}$$

$$\mathrm{SPL} = \frac{100}{N}\sum_{i=1}^{N} S_i \frac{\ell_i^\star}{\max(\ell_i, \ell_i^\star)}\%, \tag{9}$$

$$\mathrm{CR} = 100\frac{N_{\mathrm{CR}}}{N}\%, \tag{10}$$

$$\mathrm{SFR} = 100\frac{N_{\mathrm{SFR}}}{N}\%, \tag{11}$$

$$\mathrm{TLR} = 100\frac{N_{\mathrm{TLR}}}{N}\%. \tag{12}$$

Because every evaluation episode satisfies $\ell_i^\star > 0$, the SPL calculation does not require a special case for zero-length reference paths. The mutually exclusive terminal labels satisfy

$$N_{\mathrm{SR}} + N_{\mathrm{CR}} + N_{\mathrm{SFR}} + N_{\mathrm{TLR}} = N, \tag{13}$$

and therefore

$$\mathrm{SR} + \mathrm{CR} + \mathrm{SFR} + \mathrm{TLR} = 100\%, \tag{14}$$

up to numerical rounding.

All navigation metrics are micro-averaged over the corresponding evaluation episodes. Every episode contributes equal weight, irrespective of its scene, destination, goal modality, or path-length category. We do not first average within individual scenes or goal modalities. Position errors, heading errors, and action MSE retain their stated physical or loss units. Failure rates characterize the terminal outcomes.

Stop Failures at Alternative Destinations. Stop Failure at Alternative Places (SFAP) measures the proportion of stop failures that occur near a predefined but incorrect destination. Among episodes labeled as stop failures, an episode is counted as an alternative-destination stop if its terminal position lies within a planar Euclidean distance of $1.5\,\mathrm{m}$ from at least one incorrect destination:

$$\mathrm{SFAP} = 100\frac{N_{\mathrm{alternative}}}{N_{\mathrm{SFR}}}\%. \tag{15}$$

The evaluator does not assign the terminal position to the nearest destination. If the terminal position lies within $1.5\,\mathrm{m}$ of multiple incorrect destinations, the episode contributes exactly once to $N_{\mathrm{alternative}}$. Because success is evaluated before stop failure, an agent that stops within the success radius of the intended destination is counted as successful rather than as an alternative-destination failure.

SFAP is conditional on the occurrence of a stop failure and is not an additional terminal outcome. It should therefore be interpreted together with SFR. Lower SFAP indicates fewer alternative-destination confusions among the observed stop failures. If $N_{\mathrm{SFR}} = 0$, SFAP is undefined and is reported as — rather than zero.

Localization Metrics. Given a predicted pose $\hat{p}_i = (\hat{x}_i, \hat{y}_i, \hat{\theta}_i)$ and its ground-truth pose $p_i = (x_i, y_i, \theta_i)$, the position and yaw errors are

$$e_i^{\mathrm{pos}} = \left\|(\hat{x}_i, \hat{y}_i) - (x_i, y_i)\right\|_2, \tag{16}$$

$$e_i^{\mathrm{yaw}} = \left|\mathrm{wrap}(\hat{\theta}_i - \theta_i)\right|. \tag{17}$$

The position error is the planar Euclidean distance, while $\mathrm{wrap}(\cdot)$ maps the angular difference to the principal heading interval. Localization statistics are aggregated over individual localization records, with every record contributing equal weight.

B Experimental Settings and Training Ablations

B.1 Settings of the Main-Paper Experiments

Navigation and Explicit-Memory Comparisons. Tables 1 and 2 use the benchmark episodes and metrics defined in Appendix A. The former reports each goal modality separately, while the latter pools episodes across the three modalities. Adaptation interfaces, external information, and executors are specified for each baseline in Appendix C.

Pretraining Priors and Trajectory-Only Spatial Memory. Table 3 reports text-goal navigation for every variant, because the compared policies share the interface of navigating to a named destination. All variants use the same sparse post-training trajectories. The Qwen3-VL, Qwen3-VL-VLN, and Qwen3-VL-OG variants differ in whether and how navigation pretraining is applied: no navigation pretraining, VLN-format pretraining, or pretraining in the object-goal format, respectively. SMLA-Nav uses the point-goal prior and localization-based memory.

In this trajectory-only setting, every trajectory frame is also a posed localization sample for SMLA-Nav, so both localization and action supervision come from the same collected trajectories. No additional localization-grid observations are used. Thus, matched trajectory exposure does not imply identical losses: the comparison evaluates how each formulation uses the available observations, poses, and action sequences. The start-distance analysis in Appendix D.4 further partitions the text-goal results of this SMLA-Nav variant.

Stage-Wise Training and Action-Error Evaluation. Table 4 compares the pretrained policy with post-trained variants with and without scene-specific navigation trajectories. The variant without scene trajectories retains 80,000 steps from the pretraining dataset for action supervision while learning localization from scene observations. The variant with scene trajectories uses the sparse scene-specific demonstrations to adapt navigation. For each post-trained model, ground-truth and predicted absolute poses are evaluated separately, with the relative pose computed by the same deterministic adapter. These conditions separate retention of the action prior from the effect of localization error during navigation.

Table 5 uses 50,000 selected points from the evaluation trajectories. At the same points, we compare action predictions conditioned on ground-truth or predicted absolute poses against the reference motion sequences. Localization errors are measured in position and heading, and action MSE measures their downstream effect. This is an action-prediction diagnostic. Table 4 reports the corresponding closed-loop navigation evaluation.

Localization Components. The localization ablation in Fig. 6 and the localizer comparison in Table 9 both use the largest SMLA-Gauss scene, which provides a challenging setting for scene-specific localization. The figure varies localization-grid spacing and progressively adds four-view observations, rotation augmentation, and contrastive permutation to the single-view configuration. The table compares SMLA-Nav with specialized localizers using joint recall under a position threshold of $1\,\mathrm{m}$ and a heading threshold of 15°.

Contrastive Permutation. Given n localization records $\mathcal{D} = \{(V_i, p_i)\}_{i=1}^{n}$ from one scene, we generate two permutations σ and ρ of $\{1, \ldots, n\}$ and pair them element-wise:

$$\mathcal{D}_{\mathrm{CP}} = \left\{ \left((V_{\sigma(i)}, V_{\rho(i)}), (p_{\sigma(i)}, p_{\rho(i)}) \right) \right\}_{i=1}^{n}. \tag{18}$$

For example, permutations $(1, 2, 3, 4, 5)$ and $(2, 4, 5, 3, 1)$ produce paired inputs $(1, 2), (2, 4), (3, 5), (4, 3), (5, 1)$. The two observation slots represent the goal and the agent, and the shared model predicts their absolute poses in the corresponding order. Both pose fields are supervised using the existing absolute-pose token loss; no additional contrastive loss is used.

This construction produces n paired examples rather than enumerating all possible observation pairs. The underlying localization observations and pose annotations remain unchanged, while each observation appears once in each role, resulting in $2n$ pose predictions from n paired inputs. The same paired formulation is applied during localization evaluation, where localization errors are computed over the resulting pose predictions from both roles.

This formulation aligns localization supervision with the goal–agent pose prediction structure used by navigation, particularly image-goal navigation where both the destination and agent views are provided.

B.2 Absolute-Pose and Relative-Goal Fields in Point-Goal Pretraining

Experimental Design. We isolate the effect of pose conditioning by comparing three prompt configurations: relative pose only, absolute poses with the relative pose, and absolute poses only. All configurations share the same backbone, trajectories, visual inputs, action supervision, and training schedule. Ground-truth absolute poses are used, and relative poses, when provided, are computed by the geometric adapter, thereby excluding visual-localization error from this comparison. Table 7 reports the resulting point-goal navigation performance.

Results. Relative-only and absolute-plus-relative conditioning achieve nearly identical performance on both benchmarks, showing that absolute-pose fields do not impair the point-goal action interface. Absolute-only conditioning is substantially weaker, likely because the same relative pose can correspond to very different absolute coordinates across scenes and may require extrapolation beyond the pretraining range. Our absolute-plus-relative pretraining formulation therefore preserves effective point-goal navigation while remaining compatible with absolute-pose localization during scene-specific post-training.

Table 7: **Absolute-Pose and Relative-Goal Fields in Point-Goal Pretraining.** Ground-truth absolute poses are used. The absolute-plus-relative row repeats the pretraining results in Table 4.

Prompt Fields	Abs. Poses	Rel. Goal	SMLA-HM3D					SMLA-Gauss				
			SR↑	CR↓	SFR↓	TLR↓	SPL↑	SR↑	CR↓	SFR↓	TLR↓	SPL↑
Relative only	×	✓	**79.3**	19.2	1.5	0.0	**70.7**	<u>77.9</u>	16.5	4.9	0.7	**71.4**
Absolute + Relative	✓	✓	<u>79.2</u>	19.8	1.0	0.0	**70.7**	**78.1**	16.7	4.8	0.4	<u>71.2</u>
Absolute only	✓	×	68.3	25.7	2.4	3.6	54.7	69.6	23.2	3.6	3.6	59.2

B.3 Navigation Performance across Model Scales and Adaptation Strategies

Experimental Setup. We compare full-model fine-tuning and LoRA adaptation on the 8B and 2B backbones. For each backbone, a single model is jointly adapted across all scenes, with scene identity included in the prompt. LoRA uses rank $r = 64$, scaling factor $\alpha = 128$, and dropout 0.05, and is applied to the `q_proj`, `k_proj`, `v_proj`, `o_proj`, and `lm_head` modules. The adapter is merged into the backbone for inference. Point-goal pretraining is evaluated using ground-truth poses and computed relative poses. Post-training results aggregate all episodes across text, image, and coordinate goals.

Results and Analysis. As shown in Table 8, under the same point-goal protocol, the 2B backbone performs close to the 8B backbone, although 8B generally achieves higher SPL. The best 2B configuration remains close to the main 8B model, showing that the smaller backbone supports effective navigation across the 56 jointly adapted scenes.

LoRA has different effects across model sizes. For 2B, it increases SR from 78.6% to 84.8% on SMLA-HM3D and from 80.5% to 84.0% on SMLA-Gauss, while also substantially improving SPL. The constrained updates may help the smaller backbone preserve its point-goal navigation prior while learning scene-specific localization and goal grounding. LoRA also enables lightweight distribution of spatial memory: when the shared backbone is already deployed, only the compact adapter weights need to be delivered and merged for inference. For 8B, full-model post-training performs better, suggesting that the larger backbone benefits from its additional adaptation capacity. We therefore use the fully fine-tuned 8B backbone as the main SMLA-Nav model. Because each configuration is based on a single training run, the scale-dependent effect of LoRA should be interpreted cautiously.

Table 8: **Navigation Performance across Model Scales and Adaptation Strategies.**

Backbone	Stage / Adaptation	SMLA-HM3D					SMLA-Gauss				
		SR↑	CR↓	SFR↓	TLR↓	SPL↑	SR↑	CR↓	SFR↓	TLR↓	SPL↑
8B	Point-goal pretraining	79.2	19.8	1.0	0.0	70.7	78.1	16.7	4.8	0.4	71.2
	Full post-training	**86.9**	2.9	8.6	1.7	**65.6**	**85.7**	10.1	4.2	0.1	**78.4**
	LoRA post-training	80.0	3.2	11.9	4.9	61.5	85.1	12.0	2.8	0.2	78.2
2B	Point-goal pretraining	75.9	22.9	1.2	0.0	64.8	78.4	12.8	4.9	3.9	67.5
	Full post-training	78.6	3.3	11.2	6.9	56.3	80.5	15.5	3.4	0.7	68.9
	LoRA post-training	**84.8**	2.8	9.3	3.1	**65.3**	**84.0**	11.0	4.0	0.9	**75.3**

C Baseline Adaptation and Evaluation Protocols

C.1 Navigation Baselines across Goal Modalities

C.1.1 Baseline Adaptation and Evaluation

The compared methods differ in their inputs, action interfaces, scene adaptation mechanisms, and low-level controllers. We therefore adapt each baseline through the closest interface supported by its released system. Table 1 should be interpreted as a system-level comparison under these stated conditions. The controlled comparison in Table 3 separately evaluates different pretraining priors and spatial memory formulations under matched trajectory exposure.

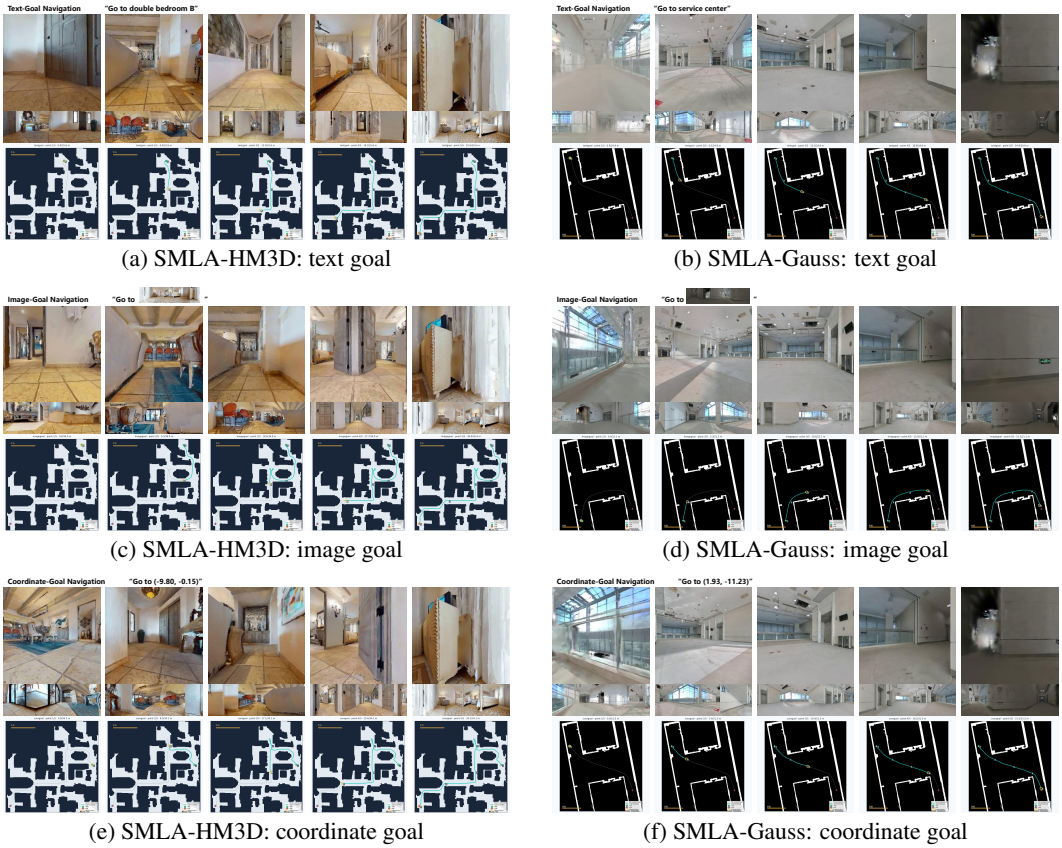
Text-Goal Navigation. Text goals use the scene-specific destination names defined in Appendix A.1. Names such as "Danny's desk" or "Pegasus meeting room" require a scene-specific name-to-location association that is not supplied by generic object semantics alone. Consequently, released zero-shot settings have a different information boundary from scene-adapted policies in this benchmark.

AstraNav predicts frontier points or a `<stop>` decision, while low-level motion is handled by an external Habitat planner. It therefore does not directly generate the collision-aware trajectories evaluated for the other methods. Because an equivalent planner is unavailable in SMLA-Gauss, we evaluate its frontier transitions and stopping decisions without assigning collision failures to AstraNav. Its zero CR consequently reflects this interface difference rather than demonstrated collision avoidance, and its results should be interpreted separately from those of trajectory-generating policies.

NavGSim and NavFoM support trajectory supervision and are therefore fine-tuned using the same sparse scene trajectories as SMLA-Nav, with the object-goal-style prompt illustrated in Fig. 5. These experiments measure adaptation under the available sparse trajectory supervision and the implemented action interfaces. Differences in backbone capacity, navigation pretraining, action parameterization, and predicted step length may also affect adaptation. The comparison does not isolate these factors. The matched-data study in Table 3 more directly examines the effects of pretraining priors and pose-based spatial supervision.

Image-Goal Navigation. UniGoal and T^2Nav are retained in their released zero-shot settings because their interfaces do not support the same scene-specific localization supervision used by SMLA-Nav. These methods rely on online exploration, observation history, and visual matching rather than acquiring persistent scene knowledge through post-training. Their results therefore measure how released zero-shot systems transfer to our location-specific evaluation setting.

Our evaluator terminates an episode immediately after the first collision. This protocol is particularly challenging in SMLA-Gauss, whose scanned scenes contain narrower passages and more complex local geometry than the simulated SMLA-HM3D environments. Correspondingly, both zero-shot image-goal baselines exhibit substantially higher collision rates on SMLA-Gauss. The observed results should therefore be interpreted together with their goal interfaces, scene exposure, and the collision protocol.



(a) SMLA-HM3D: text goal

(b) SMLA-Gauss: text goal

(c) SMLA-HM3D: image goal

(d) SMLA-Gauss: image goal

(e) SMLA-HM3D: coordinate goal

(f) SMLA-Gauss: coordinate goal

Figure 8: **Qualitative Simulation Results.** SMLA-Nav navigation examples for text, image, and coordinate goals. The left and right columns show SMLA-HM3D and SMLA-Gauss, respectively.

Coordinate-Goal Navigation. For coordinate goals, we compare with X-NavDP and MM-Nav through their point-goal interfaces. SMLA-Bench specifies the destination in world coordinates, whereas both baselines require the ground-truth agent-centric relative pose at each decision step. We therefore convert the benchmark goal into the required relative coordinate before supplying it to either method. X-NavDP additionally receives RGB and aligned depth observations and predicts executable waypoints through its released controller. These settings bypass absolute localization and coordinate-based goal grounding, so the comparisons mainly evaluate the baselines' point-goal control capabilities.

The SMLA-Nav results in Table 1 use the post-trained model, which localizes the agent, uses the supplied destination coordinate, computes the relative pose, and predicts actions. Despite performing this complete reasoning process, SMLA-Nav achieves higher SR and SPL than both X-NavDP and MM-Nav. Table 4 further shows that the pretrained SMLA-Nav point-goal policy already outperforms both baselines under ground-truth pose and relative-goal inputs. It achieves 79.2% SR and 70.7% SPL on SMLA-HM3D, and 78.1% SR and 71.2% SPL on SMLA-Gauss. Thus, the advantage is already present in the transferable point-goal navigation prior and is further combined with scene-specific localization and goal grounding during post-training.

C.1.2 Qualitative Navigation Examples

Figure 8 shows representative navigation episodes for text, image, and coordinate goals in SMLA-HM3D and SMLA-Gauss. These examples complement the aggregate results by showing the executed routes under each goal modality.

C.2 Preconstructed Explicit-Memory Baselines

We evaluate the modular systems on the same frozen episodes as SMLA-Nav: 5,264 episodes from SMLA-HM3D and 1,222 episodes from SMLA-Gauss. Results pool text, image, and coordinate goals using episode-micro aggregation. The success radii are $1.0\,\mathrm{m}$ for SMLA-HM3D and $1.5\,\mathrm{m}$ for SMLA-Gauss.

Baselines that construct scene representations receive the same scene-specific RGB observations, collection poses, and sparse trajectories available during SMLA-Nav post-training. No evaluation reference trajectory or evaluation pose label is used to construct spatial memory. As in SMLA-Nav, observed locations may overlap with later evaluation routes, but evaluation observations and pose annotations are not added to the memory-construction data.

For the reference-map pipelines, image-goal poses are estimated from the target observations without access to their ground-truth poses. Text-goal destinations are resolved through a fixed landmark registry constructed from the scene-adaptation data, while coordinate goals use the supplied world coordinate. In every task, the system estimates the robot's current pose from online visual observations.

C.2.1 Localization-Based Navigation and Component Comparisons

Reference-Map Planning and Execution. The upper block of Table 2 pairs three localization front ends—HLoc, RAVEN, and SMLA-Nav—with A^* and two execution modes: ideal polyline tracking and the post-trained SMLA-Nav executor. HLoc combines image retrieval, local feature matching, and geometric verification. RAVEN retrieves reference observations and fuses pose estimates from the four current views. Both construct their localization databases from the shared scene-specific observations and collection poses. The SMLA-Nav front end instead predicts poses using its learned parametric spatial memory.

At each decision cycle, the estimated agent position and the resolved destination position are passed to an A^* planner operating on the evaluator reference occupancy map. All localization-based configurations follow the same fallback procedure before planning. If a localization module does not return a valid pose estimate, the most recent valid estimate is reused. If the resulting planning inputs do not yield a feasible route, a deterministic rule selects a traversable candidate point to restore route feasibility before route search. These operations ensure that localization or planning issues do not terminate an episode prematurely, while success is still evaluated against the original destination. These configurations therefore evaluate navigation with an externally supplied map rather than the construction of that map.

The ideal executor directly reproduces the short motion segment prescribed by the planned polyline, without learned action prediction. Simulator pose is used only within the low-level execution bridge and is not supplied to the visual localizer or the A^* planner. For learned execution, we select a point approximately $5\,\mathrm{m}$ ahead on the planned route as the local point goal. Its position is expressed in the robot coordinate frame using the estimated current pose and supplied to the SMLA-Nav executor together with the current RGB observations. The executor predicts a motion sequence toward this local target rather than directly reproducing the full planned polyline.

Both execution modes advance for four short motion steps, corresponding to approximately $0.8\,\mathrm{m}$, before obtaining new observations, localizing again, and replanning. Thus, the $5\,\mathrm{m}$ look-ahead specifies the learned executor's target, not the distance executed between observations. Both modes operate in a repeated closed loop rather than executing the entire route without further observations.

Standalone Localization. We compare localizers in the largest SMLA-Gauss scene using joint recall under position and heading thresholds of $1\,\mathrm{m}$ and 15°, respectively. As shown in Table 9, SMLA-Nav achieves 92.28% recall, compared with 95.61% for four-view HLoc fusion and 82.11% for four-view RAVEN. Single-view HLoc achieves 83.16%. These results characterize the localization capability learned within SMLA-Nav alongside navigation. Because this comparison covers one scene, we assess benchmark-wide navigation separately through the component comparisons below.

Table 9: **Visual Localization Comparison.** Evaluation uses the largest SMLA-Gauss scene. Recall is the percentage of localization predictions satisfying both a position error of at most $1\,\mathrm{m}$ and a heading error of at most 15°.

Method	View Setting	Recall (%) ↑
HLoc (Sarlin et al., 2019)	Single view	83.16
HLoc (Sarlin et al., 2019)	Four-view fusion	**95.61**
RAVEN (Hu et al., 2026)	Four views	82.11
SMLA-Nav	Four views	92.28

Table 10: **Additional Local-Executor Comparisons with Reference-Map Pipelines.** All configurations use A^* with the evaluator occupancy map.

Localizer / Planner	Local Executor	SMLA-HM3D					SMLA-Gauss				
		SR↑	CR↓	SFR↓	TLR↓	SPL↑	SR↑	CR↓	SFR↓	TLR↓	SPL↑
HLoc (Sarlin et al., 2019) + A^*	X-NavDP (Yang et al., 2026a)	49.2	50.6	0.2	0.0	47.7	53.0	42.3	4.8	0.0	50.1
HLoc (Sarlin et al., 2019) + A^*	SMLA-Nav (pretraining)	**67.5**	26.1	6.5	0.0	**63.6**	**85.4**	10.6	4.0	0.0	**81.7**
RAVEN (Hu et al., 2026) + A^*	X-NavDP (Yang et al., 2026a)	39.1	52.4	8.6	0.0	37.6	50.4	42.8	6.8	0.0	47.7
RAVEN (Hu et al., 2026) + A^*	SMLA-Nav (pretraining)	**58.6**	27.5	13.8	0.0	**55.2**	**80.4**	13.8	5.8	0.0	**76.1**

Additional Executor and Checkpoint Comparisons. Table 10 further evaluates local execution under the reference-map interface by comparing X-NavDP and different SMLA-Nav checkpoints. X-NavDP receives the local point goal together with RGB and aligned depth observations and predicts executable waypoints. The pretrained and post-trained SMLA-Nav executors share the same point-goal interface and follow the four-step execution and replanning schedule described above.

Local execution substantially affects navigation performance. With HLoc and A^* fixed, the post-trained SMLA-Nav executor achieves 77.9% and 82.6% SR on SMLA-HM3D and SMLA-Gauss, compared with 49.2% and 53.0% for X-NavDP. The same trend is observed with RAVEN, where SR improves from 39.1% to 62.4% and from 50.4% to 77.3%. Moreover, X-NavDP performs worse than ideal polyline tracking, suggesting that waypoint prediction errors can introduce additional failures before the next replanning step. In contrast, the SMLA-Nav executor achieves higher SR than ideal tracking with both localizers (Table 2), showing the benefit of visually conditioned point-goal execution under this protocol. These comparisons involve different execution interfaces: ideal tracking reproduces the planned motion segment, whereas the learned executor predicts actions from a local goal and current visual observations.

The effect of scene-specific post-training depends on the evaluation setting. On SMLA-HM3D, post-training improves SR from 67.5% to 77.9% with HLoc and from 58.6% to 62.4% with RAVEN. However, on SMLA-Gauss, the pretrained executor achieves higher SR, with 85.4% and 80.4% compared with 82.6% and 77.3% after post-training. This indicates that scene-specific adaptation does not uniformly improve execution when the local goal is externally generated. The main-paper comparison therefore uses the post-trained checkpoint consistent with the full SMLA-Nav policy, while this analysis reports additional executor choices under the reference-map interface.

Localization Error and Execution Robustness. The performance gap between ideal tracking and learned execution also reveals the interaction between localization accuracy and local action prediction. HLoc achieves the highest localization recall among the evaluated methods, reducing the chance that the planned route starts from an incorrect map position. RAVEN retrieves poses from scene observations with known scene coordinates, although its localization accuracy is lower than HLoc.

In contrast, SMLA-Nav localization is learned jointly with navigation and may produce larger pose deviations in some cases. When combined with ideal polyline tracking, such deviations can directly affect the planned route, causing the executor to follow a motion segment generated from an inaccurate starting estimate. The learned SMLA-Nav executor alleviates this issue by conditioning action prediction on current visual observations and a local goal, rather than strictly reproducing the planned polyline. This suggests that jointly learned localization and action prediction can provide complementary robustness to intermediate pose uncertainty under the evaluated setting.

Table 11: **Effect of Replacing SMLA-Nav Localization.** HLoc and RAVEN provide current and goal poses where visual localization is required. Relative-goal computation and action prediction remain unchanged, without A^* planning.

Localization	SMLA-HM3D					SMLA-Gauss				
	SR↑	CR↓	SFR↓	TLR↓	SPL↑	SR↑	CR↓	SFR↓	TLR↓	SPL↑
SMLA-Nav	<u>86.93</u>	2.85	8.55	1.67	<u>65.58</u>	<u>85.68</u>	10.07	4.17	0.08	<u>78.40</u>
HLoc	**91.76**	2.43	5.57	0.25	**75.67**	**86.33**	8.35	5.32	0.00	**82.52**
RAVEN	83.68	3.61	10.71	1.99	63.17	82.73	12.44	4.75	0.08	75.68

Replacing SMLA-Nav Localization. We replace SMLA-Nav's current- and goal-pose predictions with HLoc or RAVEN estimates where visual localization is required. Other goal handling, relative-goal computation, and action prediction remain unchanged. These configurations condition navigation on the final destination without A^* or map-generated intermediate waypoints.

As shown in Table 11, HLoc increases SR from 86.93% to 91.76% on SMLA-HM3D and from 85.68% to 86.33% on SMLA-Gauss, while RAVEN achieves 83.68% and 82.73%, respectively. Compared with the corresponding reference-map configurations in Table 2, direct destination conditioning achieves higher SR with all three localization front ends when using the post-trained SMLA-Nav action model. This trend supports the destination-conditioned navigation formulation and is consistent with combining a transferable point-goal prior with scene-layout and route information learned from sparse trajectories.

This interpretation aligns with our use of complementary image–pose observations and sparse trajectories for scene adaptation. The common absolute-pose, relative-goal, and action format connects scene-specific spatial grounding with the pretrained navigation interface. Changing the pose provider preserves this interface and the scene-adapted action model, allowing the learned navigation policy to use either internal or external localization. HLoc's gains therefore support this composition while identifying room to improve the internal localizer. Unlike external-localizer variants that rely on scene-specific localization databases constructed from stored observations, the full SMLA-Nav policy encodes scene memory in model parameters and does not query external scene representations at inference. This removes the need to maintain a separate localization database while preserving destination-conditioned navigation.

Reference-Map Composition and Direct Destination Conditioning. Taken together, Tables 2 and 11 show that navigation performance depends on how pose estimates are used to condition action prediction, in addition to the choice of pose provider. With HLoc, direct destination conditioning achieves 91.76% and 86.33% SR, compared with 77.94% and 82.57% for HLoc+A^* with the post-trained SMLA-Nav executor. With RAVEN, the corresponding comparison is 83.68% and 82.73% versus 62.37% and 77.25%. The higher SR with both external localizers supports the effectiveness of destination-conditioned visual navigation under the evaluated settings. These are comparisons of complete navigation interfaces: route generation and goal conditioning change together, rather than only the presence of an A^* call.

The learned components also support effective navigation when combined with explicit map planning. With nearest-traversable-point projection, SMLA-Nav localization coupled with A^* and the post-trained executor achieves 85.37% and 83.47% SR. The full policy achieves 86.93% and 85.68% without querying the reference map. Their SPL scores are nearly identical on SMLA-HM3D (65.59% for the reference-map combination and 65.58% for the full policy), while the full policy obtains higher SPL on SMLA-Gauss (78.40% versus 76.69%). Thus, the learned localization and action components are useful both in an explicit-map composition and in the full policy, which does not require map-based route search.

Reference-map configurations nevertheless remain competitive. In particular, HLoc with the pretrained SMLA-Nav executor achieves 85.4% SR and 81.7% SPL on SMLA-Gauss, compared with 85.7% and 78.4% for the full policy. The complete SMLA-Nav model therefore provides effective navigation without external scene-memory queries, rather than uniformly dominating every component configuration on both SR and SPL.

Table 12: **Navigation with Single-View Observations.** The upper panel evaluates point-goal pretraining with a single forward view. The lower panel reports navigation after scene-specific post-training with single-view localization.

Goal	SMLA-HM3D					SMLA-Gauss				
	SR↑	CR↓	SFR↓	TLR↓	SPL↑	SR↑	CR↓	SFR↓	TLR↓	SPL↑
Navigation pretraining with a single forward view										
Point Goal	78.2	16.8	5.0	0.0	70.3	78.9	7.8	13.3	0.0	71.7
Scene-specific post-training with single-view localization										
Text Goal	41.14	2.96	54.71	1.18	31.63	52.0	18.5	25.9	3.6	43.0
Image Goal	16.32	5.26	76.08	2.33	11.87	26.9	30.2	38.1	4.8	21.0
Coordinate Goal	45.87	3.07	50.29	0.77	35.60	43.5	24.4	30.2	1.8	35.2
Overall	33.85	3.80	60.89	1.46	25.88	40.9	24.4	31.3	3.4	33.1

C.2.2 Memories Constructed from Scene Data

Construction and Adaptation. ULVN, OpenNavMap, and MASt3R-Nav construct spatial representations before evaluation from the shared scene-specific data. ULVN combines a visual topological representation with ViNT execution. OpenNavMap uses a traversability representation with iPlanner. MASt3R-Nav constructs a pixel-level spatial representation executed by PixelReact.

We adapt their goal interfaces to the three SMLA-Bench modalities and replace any ground-truth reference-frame selection with online visual localization to avoid test-pose leakage. These systems are controlled adaptations to our evaluation setting rather than exact reproductions of their original task protocols. Unlike the reference-map configurations, they construct their scene representations from the supplied scene data rather than receiving the evaluator occupancy map.

Results and Analysis. As shown in Table 2, SMLA-Nav achieves 86.9% and 85.7% SR on SMLA-HM3D and SMLA-Gauss, respectively, compared with 50.8% and 55.7% for OpenNavMap, the strongest of the three constructed-memory baselines. The corresponding SPL scores improve from 43.8% to 65.6% and from 47.8% to 78.4%. Under matched scene-data access, these results support the effectiveness of learning reusable scene knowledge and composing it with navigation action prediction within a shared model, without querying a constructed scene representation at inference.

The comparison evaluates complete systems, whose memory representations and execution components differ. It therefore supports SMLA-Nav's learning-and-composition scheme under the stated adaptations and data budget, rather than an intrinsic superiority of implicit over explicit spatial memory.

D Robustness and Generalization Analyses

D.1 Effect of View Configuration

We separately examine the effect of view configuration on navigation pretraining and scene-specific localization. As shown in Table 12, point-goal pretraining with a single forward view performs comparably to the default four-view setting reported in Table 4. This suggests that a single view is generally sufficient for learning the navigation prior. In contrast, replacing four-view localization with single-view localization substantially reduces post-training navigation performance on both benchmarks. Together with the localization ablation in Fig. 6, this contrast supports using four views to reduce visual ambiguity during pose estimation. Image-goal navigation is affected most, consistent with its need to localize both the current observation and the goal image before computing their relative displacement. Errors from either estimate can affect action prediction.

D.2 Evaluation with Noise in Post-Training Pose Labels

To evaluate sensitivity to imperfect pose annotations during scene adaptation, we add zero-mean Gaussian noise to the pose labels in the post-training data. We consider position and heading standard deviations of $(0.2\,\mathrm{m}, 15^\circ)$ and $(0.4\,\mathrm{m}, 30^\circ)$. Table 13 reports the results for each goal modality

Table 13: **Robustness to Gaussian Noise in Post-Training Pose Labels.** Noise levels denote the standard deviations of position and heading perturbations. Displayed terminal rates may differ slightly from 100% due to rounding.

Pose Noise $(\sigma_p, \sigma_\theta)$	Task	SMLA-HM3D					SMLA-Gauss				
		SR↑	CR↓	SFR↓	TLR↓	SPL↑	SR↑	CR↓	SFR↓	TLR↓	SPL↑
$(0.2\,\mathrm{m}, 15°)$	Text	78.4	3.4	13.1	5.1	56.0	81.2	6.9	10.4	1.5	72.2
	Image	74.7	4.5	15.6	5.2	53.6	75.9	12.7	10.4	1.0	62.6
	Coordinate	81.1	3.3	12.3	3.3	58.5	83.6	9.2	6.0	1.2	70.9
	Overall	77.9	3.7	13.8	4.6	55.9	80.4	9.6	8.8	1.2	68.6
$(0.4\,\mathrm{m}, 30°)$	Text	73.6	3.1	17.7	5.6	51.2	68.5	12.2	14.0	5.3	58.0
	Image	65.5	3.1	23.9	7.5	44.6	62.2	17.0	16.5	4.3	49.5
	Coordinate	73.0	3.5	18.4	5.1	51.3	64.3	15.7	16.4	3.7	53.0
	Overall	70.6	3.2	20.1	6.1	48.9	65.0	15.0	15.6	4.4	53.5

and their episode-level aggregate. The corresponding clean results are reported in Table 1. Under the milder noise, the overall SR decreases from 86.9% to 77.9% on SMLA-HM3D and from 85.7% to 80.4% on SMLA-Gauss. With the stronger perturbation, the overall SR remains 70.6% and 65.0%, respectively, while every goal modality retains an SR above 62%. These results show that SMLA-Nav retains part of its navigation performance under the evaluated levels of post-training pose-label noise. This experiment characterizes sensitivity to training-label noise rather than online localization errors or accumulated odometric drift.

D.3 Effect of Localization Grid Spacing

To examine how the density of localization supervision affects scene-specific spatial memory, we increase the localization-grid spacing from the default $0.5\,\mathrm{m}$ to $1.0\,\mathrm{m}$ and $2.0\,\mathrm{m}$ while keeping the navigation trajectories and evaluation episodes unchanged. As shown in Table 14, navigation SR decreases as the localization observations become sparser. Increasing the grid spacing to $1.0\,\mathrm{m}$ reduces the overall SR from 86.9% to 76.2% on SMLA-HM3D and from 85.7% to 75.1% on SMLA-Gauss. With the substantially sparser $2.0\,\mathrm{m}$ grid, the model still achieves overall SRs of 65.6% and 64.8%, respectively. SPL exhibits a similar progressive decrease, while all three goal modalities retain an SR above 61% under the sparsest setting. The consistent trend across both benchmarks, without a disproportionate collapse in any goal modality, indicates that SMLA-Nav can trade localization-data density for navigation performance and remains effective with fewer scene-specific localization observations.

Table 14: **Effect of Localization Grid Spacing.** The default $0.5\,\mathrm{m}$ results are reported in Table 1. Displayed terminal rates may differ slightly from 100% due to rounding.

Grid Spacing	Task	SMLA-HM3D					SMLA-Gauss				
		SR↑	CR↓	SFR↓	TLR↓	SPL↑	SR↑	CR↓	SFR↓	TLR↓	SPL↑
$1.0\,\mathrm{m}$	Text	77.9	5.4	13.6	3.1	59.7	75.9	11.7	11.2	1.3	71.3
	Image	74.3	3.8	17.5	4.4	55.0	70.8	22.1	6.3	0.8	63.4
	Coordinate	76.4	5.9	14.9	2.8	58.1	78.3	14.5	6.9	0.2	70.6
	Overall	76.2	5.0	15.4	3.5	57.6	75.1	16.0	8.1	0.7	68.5
$2.0\,\mathrm{m}$	Text	67.1	8.1	20.2	4.6	51.4	65.5	16.5	16.2	1.8	61.4
	Image	64.0	5.3	24.6	6.1	47.3	61.2	29.7	8.4	0.8	54.6
	Coordinate	65.8	8.5	21.6	4.1	50.1	67.5	21.7	10.4	0.5	60.8
	Overall	65.6	7.2	22.2	5.0	49.6	64.8	22.6	11.6	1.0	59.0

D.4 Navigation from Starts Distant from Post-Training Trajectories

Table 15 expands the trajectory-only SMLA-Nav result in Table 3 using text-goal episodes. We define $d_{\mathrm{start}} = \min_{\mathbf{p}\in\mathcal{P}_{\mathrm{traj}}} \|\mathbf{p}_{\mathrm{start}} - \mathbf{p}\|_2$ as the distance from an evaluation start to its nearest

Table 15: **Text-Goal Navigation from Starts Distant from Post-Training Trajectories.** This table expands the trajectory-only SMLA-Nav result in Table 3. d_{start} denotes the Euclidean distance from the evaluation start to its nearest post-training trajectory point. N is the number of evaluation episodes.

d_{start}	SMLA-HM3D						SMLA-Gauss					
	N	SR↑	CR↓	SFR↓	TLR↓	SPL↑	N	SR↑	CR↓	SFR↓	TLR↓	SPL↑
$\leq 0.5\,\text{m}$	1,438	72.7	3.1	20.4	3.8	49.4	285	77.2	10.5	8.4	3.9	67.9
$(0.5, 1]\,\text{m}$	254	73.2	2.0	20.9	3.9	49.1	56	62.5	17.9	17.9	1.8	51.5
$(1, 2]\,\text{m}$	86	60.5	3.5	36.0	0.0	46.6	28	60.7	21.4	14.3	3.6	42.6
$(2, 4]\,\text{m}$	79	43.0	2.5	35.4	19.0	34.9	21	61.9	14.3	9.5	14.3	53.6
$> 4\,\text{m}$	0	—	—	—	—	—	4	25.0	25.0	50.0	0.0	24.0
Overall	1,857	71.0	2.9	21.8	4.3	48.6	394	72.6	12.7	10.7	4.1	62.5

Table 16: **Results by Evaluation Path Length.** Displayed outcome rates may differ from 100% by 0.1% due to independent rounding. SMLA-HM3D contains no long-path episodes.

Subset	Path length	Goal	N	SR↑	CR↓	SFR↓	TLR↓	SPL↑
SMLA-HM3D	Short: $\ell^{\star}_{\text{eval}} \leq 10\,\text{m}$	Coordinate	1,134	86.9	3.4	8.9	0.9	64.5
		Image	1,424	85.1	2.9	9.6	2.3	62.3
		Text	1,408	89.5	2.3	7.2	0.9	66.0
	Medium: $10 < \ell^{\star}_{\text{eval}} \leq 30\,\text{m}$	Coordinate	429	86.5	3.7	8.6	1.2	69.7
		Image	420	81.7	3.3	10.7	4.3	64.7
		Text	449	90.2	1.6	6.2	2.0	74.4
SMLA-Gauss	Short: $\ell^{\star}_{\text{eval}} \leq 10\,\text{m}$	Coordinate	211	87.2	8.5	4.3	0.0	74.2
		Image	182	82.4	14.3	3.3	0.0	67.4
		Text	162	92.6	2.5	4.9	0.0	86.4
	Medium: $10 < \ell^{\star}_{\text{eval}} \leq 30\,\text{m}$	Coordinate	82	87.8	9.8	2.4	0.0	81.3
		Image	88	79.5	17.0	3.4	0.0	73.7
		Text	98	82.7	15.3	2.0	0.0	78.7
	Long: $30 < \ell^{\star}_{\text{eval}} \leq 100\,\text{m}$	Coordinate	141	85.8	10.6	3.5	0.0	82.6
		Image	124	75.8	13.7	10.5	0.0	74.1
		Text	134	93.3	3.7	2.2	0.7	91.0

post-training trajectory point. Here, $\mathbf{p} \in \mathbb{R}^2$ denotes a planar position. We express SMLA-HM3D positions in the training coordinate frame $(-x, z)$ and SMLA-Gauss positions as (x, y). Heading is excluded from the distance. OOD characterizes starts farther from observed trajectory locations within an adapted scene.

Because this setting uses no additional localization grid, distant starts have no nearby pose supervision. Localization must extrapolate from visual appearance, which can bias the predicted absolute pose and the resulting relative pose. On SMLA-HM3D, SR is 72.7% within $0.5\,\text{m}$ and 73.2% at 0.5–$1\,\text{m}$, then decreases to 60.5% at 1–$2\,\text{m}$ and 43.0% at 2–$4\,\text{m}$. On SMLA-Gauss, the 1–$4\,\text{m}$ bins contain only 49 episodes, and the $> 4\,\text{m}$ bin contains four. Its more distant bins should therefore be interpreted cautiously.

Navigation remains possible for some starts outside the trajectory-covered regions of the adapted scene. At 2–$4\,\text{m}$ on SMLA-HM3D, the retained SR together with the relatively low CR suggests that the pretrained navigation prior may sometimes maintain traversable motion under uncertain initial localization. Closed-loop execution may also move the agent toward a better-covered region and enable recovery in some episodes. Overall, the results suggest partial robustness to starts distant from post-training trajectories, with performance decreasing on SMLA-HM3D and varying non-monotonically across the smaller SMLA-Gauss groups.

D.5 Results by Scene Size and Evaluation Path Length

We stratify the evaluation episodes by geodesic shortest-path length ℓ^{*}_{eval} and navigable scene area A_{scene}, using the intervals defined in Table 6. Each episode receives equal weight within its goal modality. Tables 16 and 17 report the corresponding results.

Table 17: **Results by Navigable Scene Area.** Displayed outcome rates may differ from 100% by 0.1% due to independent rounding. SMLA-HM3D contains no large scenes.

Subset	Scene area	Goal	N	SR↑	CR↓	SFR↓	TLR↓	SPL↑
SMLA-HM3D	Small: $A_{\text{scene}} \leq 50\,\text{m}^2$	Coordinate	690	86.8	3.2	9.6	0.4	66.7
		Image	823	84.6	4.0	10.0	1.5	63.3
		Text	834	89.3	2.4	7.7	0.6	68.5
	Medium: $50 < A_{\text{scene}} \leq 300\,\text{m}^2$	Coordinate	873	86.7	3.7	8.2	1.4	65.3
		Image	1,021	84.1	2.3	9.8	3.8	62.5
		Text	1,023	89.9	2.0	6.5	1.7	67.7
SMLA-Gauss	Small: $A_{\text{scene}} \leq 50\,\text{m}^2$	Coordinate	169	87.6	7.1	5.3	0.0	74.2
		Image	136	79.4	16.9	3.7	0.0	64.4
		Text	136	90.4	2.2	7.4	0.0	84.7
	Medium: $50 < A_{\text{scene}} \leq 300\,\text{m}^2$	Coordinate	35	97.1	0.0	2.9	0.0	91.3
		Image	28	96.4	0.0	3.6	0.0	91.2
		Text	28	100.0	0.0	0.0	0.0	98.0
	Large: $300 < A_{\text{scene}} \leq 800\,\text{m}^2$	Coordinate	230	84.8	12.6	2.6	0.0	79.3
		Image	230	77.8	15.2	7.0	0.0	72.3
		Text	230	89.1	9.1	1.3	0.4	85.3

Effect of Evaluation Path Length. Performance does not decrease monotonically with path length. After pooling the three goal modalities at the episode level, SR is 87.2% and 86.2% in the short- and medium-path subsets of SMLA-HM3D. On SMLA-Gauss, SR is 87.2%, 83.2%, and 85.2% for short, medium, and long paths, respectively. Long-path SPL exceeds short-path SPL for each SMLA-Gauss goal modality. At every closed-loop decision step, SMLA-Nav uses the current multi-view observation and goal to estimate its pose and recompute the relative pose. The policy uses a fixed number of current views rather than accumulating observation history as a route grows.

Effect of Scene Size. Area does not show a monotonic relationship with navigation SR. After pooling the goal modalities at the episode level, SR is 86.9% in both the small- and medium-area SMLA-HM3D groups. On SMLA-Gauss, the corresponding SR values are 85.9%, 97.8%, and 83.9% for small, medium, and large scenes, respectively. SMLA-Nav relocalizes from the current observation at each decision step, keeping the input length at each step fixed across scene sizes. This property may contribute to the observed non-monotonic trend. The medium and large SMLA-Gauss groups contain two and one scenes, respectively, so differences between these groups should not be attributed solely to scene area.

E Real-World Deployment Details

Deployment Site and Adaptation Data. We deploy SMLA-Nav on a Unitree Go2 quadruped at the approximately $86.2\,\text{m} \times 56.0\,\text{m}$ site shown in Fig. 7. Each data-collection run begins from the same marked physical position and heading, allowing observations from different runs to share a common site coordinate system. We teleoperate the robot through corridors, open areas, intersections, and visually ambiguous regions.

The collected dataset contains approximately 4,000 four-view RGB observations with absolute pose labels (x, y, θ). Approximately 3,000 of these observations additionally belong to complete navigation trajectories and therefore provide action supervision $(\Delta u, \Delta v, \Delta\alpha)$. The action labels are fitted from the collected trajectories. The fitted action sequences are then resampled so that the translational and rotational intervals between consecutive action steps match those used during pretraining, reducing the distribution gap between pretraining and real-world adaptation. The remaining observations provide absolute localization supervision only. Post-training on the collected data takes approximately one hour.

Pose Supervision. Initial poses are obtained from the Unitree SDK2 onboard state estimator, which combines inertial measurements with leg odometry. We subsequently refine the collected trajectories offline using LiDAR constraints, loop-closure constraints, and reliable global position measurements when available. After temporal alignment and quality filtering, all poses are repre-

sented in the common coordinate system defined by the shared physical start pose. Samples with large synchronization errors or unreliable pose estimates are removed before post-training.

Inference and Robot Control. At each navigation step, the high-level model receives the current four-view RGB observation and the goal specification. It predicts a sequence of actions $(\Delta u, \Delta v, \Delta\alpha)$, which is then returned to the robot for execution. The absolute pose labels used during post-training are not provided to the high-level model during evaluation. The robot's onboard controller tracks the predicted actions, while its onboard obstacle-avoidance function remains enabled as a safety layer.

Table 18: **Real-World Navigation Results at One Adapted Site.** All trials begin from the same physical start pose.

Goal Modality	Number of Trials	SR (%)
Text Goal	20	75.0
Image Goal	20	65.0
Coordinate Goal	20	80.0

Evaluation Results and Failure Analysis. As shown in Table 18, we conduct 20 trials for each goal modality. SMLA-Nav achieves success rates of 75.0%, 65.0%, and 80.0% for text, image, and coordinate goals, respectively.

Observed failures included repeated localization errors at visually similar intersections, which could direct the robot onto an incorrect branch and prevent recovery to the destination. Such episodes were counted as failures when the robot missed the destination and could not return to the correct route.

Several factors may contribute to the differences among goal modalities. Coordinate goals achieve the highest success rate, while image goals achieve the lowest. We hypothesize that coordinate goals benefit from directly specifying the target in the adapted site coordinate system, whereas image goals are more affected by viewpoint differences, illumination changes, partial occlusions, and perceptual aliasing between visually similar regions.

Qualitative Navigation Results. Figure 9 presents real-world navigation episodes with text, image, and coordinate goals. The quadruped robot reaches a named meeting room, a location specified by a goal image, and a target specified in scene coordinates, respectively. These sequences illustrate how successive short-horizon trajectory predictions support navigation through corridors and turns to destinations specified in different modalities.

Robustness and Limitations. The system remains operational in trials involving dynamic pedestrians and temporary visual occlusions. We hypothesize that pedestrian-rich adaptation data and complementary information from the four-view observations contribute to this behavior. The onboard obstacle-avoidance module also remains enabled as a safety layer during physical execution.

In contrast, the system is more sensitive to persistent changes in the static scene layout. If large facilities, barriers, booths, or other location-defining objects are moved, the visual-to-spatial correspondences learned during site adaptation may become invalid, leading to systematic localization errors. Pose-label noise, odometric drift, communication and inference latency, and interventions by the onboard obstacle-avoidance module may also affect the measured navigation performance.

Finally, all experiments are conducted at a single adapted site and begin from the same physical start pose. The reported results therefore reflect performance under this controlled deployment setting and do not yet establish robustness to major scene reconfiguration, different starting regions, or previously unseen environments.



Figure 9: **Real-world navigation examples.** Rows show navigation with text, image, and coordinate goals, from top to bottom. Each row presents successive stages from left to right, including external views of the robot, onboard observations, and trajectory visualizations. The final column shows arrival at the specified destination.

F LIMITATIONS

SMLA-Nav requires posed scene observations and remains sensitive to sparse coverage, label noise, and persistent scene changes. The robustness experiments characterize these effects within the evaluated environments.

Recomputing a fixed text or image destination at every step introduces redundant inference. Future work could cache a reliable goal pose, use coordinate conditioning, and trigger goal grounding again only when needed.

Repeated short-horizon predictions enable longer routes but do not guarantee globally optimal paths or recovery from persistent localization errors. Text goals currently use fixed landmark names. Relational descriptions and broader semantic-spatial reasoning remain untested. Real-world evaluation covers one adapted site and a shared start pose (Appendix E).