Sparse Video Generation Propels Real-World Beyond-the-View Vision-Language Navigation

Hai Zhang* **Siqi Liang*** **Li Chen** **Yuxian Li**

Yukuan Xu **Yichao Zhong** **Fu Zhang** **Hongyang Li**

The University of Hong Kong
https://github.com/OpenDriveLab/SparseVideoNav

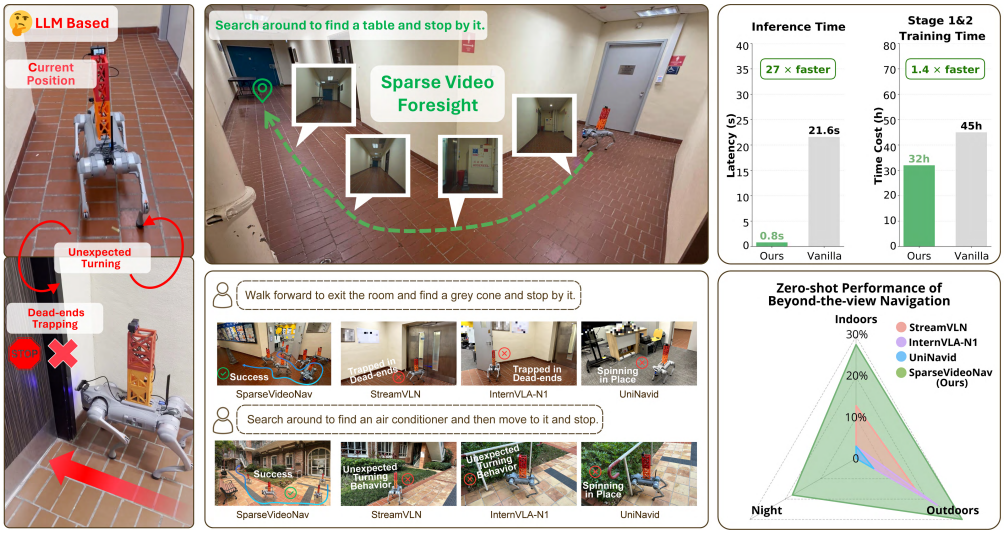

Figure 1: In this work, we investigate beyond-the-view navigation task in the real world, where agents must locate distant, unseen targets without step-by-step guidance. Traditional large language model-based methods suffer from short-horizon supervision, leading to short-sighted behaviors, *e.g.*, unexpected turning and dead-end trapping. To address this challenge, we pioneer the use of video generation model as the primary backbone for action generation. The whole training pipeline is sparsified further for the sake of extended prediction horizon and computational efficiency.

Abstract

Why must vision-language navigation be bound to detailed and verbose language instructions? While such details ease decision-making, they fundamentally contradict the goal for navigation in the real-world. Ideally, agents should possess the autonomy to navigate in unknown environments guided solely by simple and high-level intents. Realizing this ambition introduces a formidable challenge: Beyond-the-View Navigation (BVN), where agents must locate unseen targets without step-by-step guidance. Existing large language model (LLM)-based methods, though adept at following dense instructions, often suffer from short-sighted behaviors due to their reliance on short-horizon supervision. Simply extending the supervision horizon, however, destabilizes LLM training. In this work, we

*Equal contribution

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

identify that video generation models (VGM) inherently benefit from long-horizon supervision to align with language instructions, rendering them uniquely suitable for BVN tasks. Capitalizing on this insight, we propose applying VGM as the main backbone to generate actions for the first time in this field. Yet, the prohibitive latency for generating videos spanning tens of seconds makes real-world deployment impractical. To bridge this gap, we propose SparseVideoNav, achieving sub-second trajectory inference guided by a generated sparse future spanning a 20-second horizon. This yields a remarkable 27× speed-up compared to the unoptimized counterpart. Extensive real-world zero-shot experiments demonstrate that SparseVideoNav achieves 2.5× the success rate of state-of-the-art LLM baselines on BVN tasks and marks the first realization of such capability in challenging night scenes.

1 Introduction

Vision-language navigation (VLN) empowers agents to perform complex tasks by grounding natural language instructions into sequential actions based on visual observations [Ku et al., 2020a, Zhang et al., 2026a]. Recently, the advent of large-language models (LLMs) has catalyzed significant breakthroughs in this field [Zhang et al., 2025a, 2026b, Wei et al., 2026a]. However, a fundamental tension remains: while real-world interactions typically demand navigation based on simple and high-level intents, current agents often require dense and step-by-step instructions [Cheng et al., 2025, Wei et al., 2026b], a paradigm typically termed Instruction-Following Navigation (IFN). This clear discrepancy highlights a formidable but less-explored frontier: *Beyond-the-View Navigation* (BVN), where agents must autonomously locate unseen targets in the absence of such granular and intermediate guidance.

The primary bottleneck in BVN for existing LLM-based methods is their inherent short sight. These models are typically supervised by short-horizon actions (*e.g.*, 4 to 8 steps) during training [Wei et al., 2026b, Zhang et al., 2025a, Team, 2025]. Consequently, they struggle to infer correct navigational intents, typically causing two failure modes during deployment. First, the inability to observe the target over a long distance induces severe uncertainty, resulting in unexpected turning behaviors or spinning in place. Second, upon entering a dead end, the agent mistakenly assumes the path end, leading to dead-end trapping. While extending the supervision horizon seems like a logical fix, it often destabilizes the training process of LLM [Wang et al., 2025], rendering it a non-viable solution.

In this work, we pivot toward video generation models (VGMs) based on a key observation: unlike LLMs, VGMs are inherently pre-trained to capture long-horizon future aligned with the language instruction [Wan et al., 2025, Wu et al., 2025]. Recognizing this strength, we identify VGMs as a well-suited interface to provide the long-horizon foresight that BVN demands. Nevertheless, whether we should strictly adhere to the standard VGM paradigm of predicting ***continuous*** videos [Hu et al., 2025, Team, 2025] remains a question worthy of scrutiny. We contend that the dense and high-frequency temporal information required by continuity is redundant for guiding navigation. This insight drives us to explore a ***sparse*** video generation paradigm. By doing so, we aim to achieve the dual advantage of extending the prediction horizon while simultaneously reducing both training and inference overhead. To this end, we reformulate the training objective by employing sparse video frames corresponding to strategically selected timesteps as the direct supervision signal for VGM.

Translating this sparse foresight into a comprehensive navigation system, yet, presents two challenges. Primarily, the computational overhead of injecting the whole history, coupled with the extensive denoising steps required to generate future predictions, still leads to significant inference latency that makes real-world deployment impractical. In addition, unlike LLM-based methods that can bridge the sim-to-real gap by co-training with heterogeneous real-world data [Zhang et al., 2025a, Wei et al., 2026b], VGM lacks a straightforward mechanism to leverage such data sources.

To bridge this gap, we introduce SparseVideoNav, a **Sparse Video** generation-based system for vision-language **Nav**igation. For the efficiency bottleneck, we instantiate SparseVideoNav via a structured training pipeline. For the data challenge, we curate a large-scale navigation dataset spanning 140 hours across diverse real-world environments. Through zero-shot evaluations in six real-world scenes, we demonstrate that SparseVideoNav achieves state-of-the-art (SOTA) performance on BVN tasks, marking the first realization of such capability in challenging night scenes.

Contributions:

1) We investigate beyond-the-view navigation tasks in the real world from a new perspective by applying video generation model as the primary backbone for action generation. Crucially, we empirically identify that supervising with longer horizon is instrumental in improving performance.

2) We advance beyond the conventional constraint of continuous video generation in existing methods by pioneering a paradigm shift towards sparse video generation. This reformulation liberates the capability to reason over substantially longer prediction horizon while achieving $1.4\times$ training speed-up and $1.7\times$ inference speed-up.

3) SparseVideoNav achieves sub-second trajectory inference guided by a generated sparse future spanning a 20-second horizon. This yields a remarkable $27\times$ speed-up compared to the unoptimized counterpart. Extensive real-world zero-shot experiments demonstrate SparseVideoNav adeptly handles beyond-the-view tasks even in complex terrains like dead ends, narrow accessible ramp, and hillside with high inclination angles, achieving $2.5\times$ the success rate of SOTA LLM baselines and marking the first realization of such capability in challenging night scenes with a 17.5% success rate.

4) We open-source a real-world navigation dataset spanning over 140 hours, representing the largest-scale real-world navigation dataset to date.

2 Related Work

2.1 Foundation Models for Vision-Language Navigation

The integration of foundation models into VLN predominantly transitions from modular, training-free approaches to end-to-end fine-tuning approaches. Training-free approaches leverage the zero-shot reasoning capabilities of off-the-shelf LLMs to perform specific sub-tasks, such as task reasoning [Long et al., 2024], stage scheduling [Cao et al., 2025], frontier detection [Gong et al., 2025], and object validation [Yu et al., 2024]. While these modular approaches maintain interpretability, they inherently suffer from cascading error propagation [Yu et al., 2024] or limited generalization capability [Yokoyama et al., 2024, Chen et al., 2025].

To better optimize spatio-temporal information throughout the training pipeline, fine-tuning approaches have gained increasing traction, which aim to map multimodal inputs directly to actions in an end-to-end manner [Zhang et al., 2025a,b]. Recent SOTA methods have demonstrated strong generalization through large-scale training. Notably, Zhang et al. [2025a] proposes Uni-Navid to establish a competitive baseline by unifying diverse navigation tasks with a single versatile policy. StreamVLN [Wei et al., 2026b] introduces a streaming framework that manages continuous video inputs via hybrid slow-fast context modeling to accelerate the inference. Despite their promise, these methods predominantly require dense and step-by-step instructions, a dependency that contradicts with practical real-world demands. When facing beyond-the-view tasks guided by simple and high-level intents, they are prone to falter due to the vulnerability induced by short-horizon supervision. Consequently, there is a pressing need for a novel paradigm designed to bridge this gap.

2.2 Video Generation Models for Embodied Agents

Distinguished from text-centric pretraining, VGM implicitly encodes dynamic variation between frames, leading to a smaller domain gap towards downstream actions. This unique property has driven an explosion of interest across a wide spectrum of embodied tasks from robotic manipulation [Bu et al., 2024, Hu et al., 2025, Ye et al., 2026] to autonomous driving [Gao et al., 2024, Yang et al., 2025, Zhao et al., 2025]. Within the specialized scope of navigation, InternVLA-N1 [Team, 2025] demonstrates that co-training with video generation objectives can improve downstream policy performance. However, its primary backbone remains LLM, making it susceptible to the limitations of short-horizon supervision. To the best of our knowledge, SparseVideoNav is the first to apply VGM as the primary backbone for action generation in VLN. Unlike Navigation World Model [Bar et al., 2025], which is restricted to predicting the immediate next frame based on low-level action inputs, our framework is purely language-conditioned, directly processing high-level intents to achieve long-horizon video foresight. Concurrent work [Huang et al., 2026] also uses VGM as a zero-shot high-level navigator, but does not train the VGM backbone to further improve the performance.

Noticeably, these endeavors adopt the default prerequsite to generate continuous videos. While some concurrent works in manipulation explore sparsification to balance action and video tokens [Bi et al., 2026, Li et al., 2026], their future predictions are typically limited to a mere few seconds. In

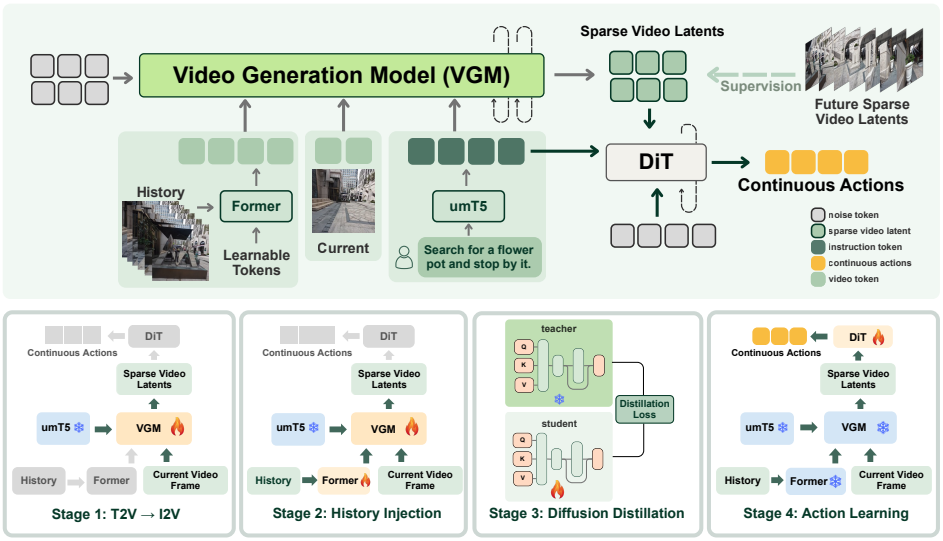

Figure 2: **Architecture and four-stage training pipeline of SparseVideoNav**. **(Top)** denotes our whole training architecture. Current observation, historical observations, and the language instruction are fed into the video generation model (VGM) backbone to generate future sparse video latents. DiT-based action head then predicts continuous actions conditioned on generated sparse future and the language instruction. **(Bottom)** denotes our four-stage training pipeline, with **Stage 1 (Sec. 3.3)** adapting T2V to I2V, **Stage 2 (Sec. 3.4)** injecting history into I2V backbone; **Stage 3 (Sec. 3.5)** distilling the backbone to reduce denoising steps; **Stage 4 (Sec. 3.6)** learning actions based on generated sparse future. Components not utilized in a specific stage are indicated by gray blocks.

contrast, we are the first to harness the sparsification for significantly extending the prediction span (*e.g.*, 20 seconds). This enables efficient long-horizon supervision, ultimately unlocking substantial performance improvements in BVN tasks.

3 Methodology

The design of SparseVideoNav is guided by two fundamental pillars: a training paradigm-level innovation through sparsification (Sec. 3.1), and a system-level innovation via synergistic orchestration. To achieve this, we develop a data curation pipeline (Sec. 3.2) alongside a structured four-stage training pipeline. The linchpin of the training pipeline is the *sparse future video*, which serves as the supervision across stages 1, 2, and 3, and the conditional input of stage 4.

The whole architecture and training pipeline are shown in Fig. 2, where `Former` block denotes Q-Former [Li et al., 2023] and Video-Former [Hu et al., 2025] used to compress history.

3.1 Sparsification

To extend the prediction horizon, we bring in sparse video supervision into the training of VGM to enable fixed-interval video generation. As shown in Fig. 3, setting the interval to 3 strikes the best balance between the prediction horizon and the visual fidelity. To further ensure action prediction accuracy, we maintain continuous generation for the first two observation chunks, covering 8 timesteps. This design leads to the sparse generation timesteps at

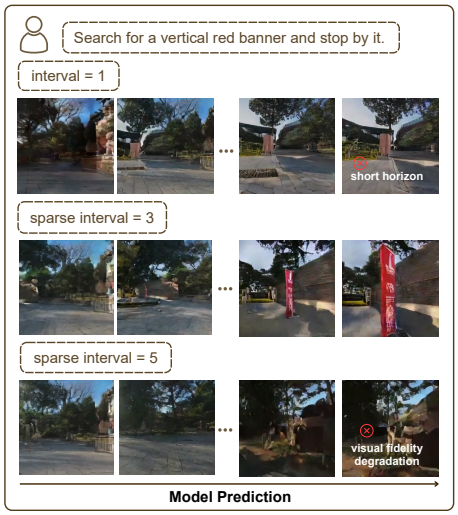

Figure 3: **Qualitative comparison of different sparse intervals.** With the sparse interval of 3, the model successfully imagines a path towards beyond-the-view target, while maintaining visual fidelity.

$[T+1, T+2, T+5, T+8, T+11, T+14, T+17, T+20]$, spanning 20 seconds at 4 FPS, where T denotes the current timestep.

3.2 Data Curation Pipeline

Different from methods built upon video-based LLMs [Zhang et al., 2025a, 2026b], which can mitigate the sim-to-real gap by co-training on pure simulation navigation data [Savva et al., 2019] mixed with real-world VQA datasets, VGM cannot effectively leverage this strategy. Relying exclusively on simulation data typically leads to mode collapse [Srivastava et al., 2017, Thanh-Tung and Tran, 2020]. Furthermore, existing real-world navigation datasets are often characterized by severe fisheye distortion [Shah et al., 2021, Hirose et al., 2023] and limited scale [Hirose et al., 2018], making them unsuitable for fine-tuning VGM.

To address these challenges, we develop a data curation pipeline involving human operators equipped with a handheld camera to capture diverse videos. To minimize human jitter that blocks VGM to learn consistent dynamics, we explicitly utilize DJI Osmo Action 4 with RockSteady+ stabilization.

We collect 140 hours of real-world navigation videos. These videos are processed into approximately 13,000 trajectories via uniform temporal sampling, with an average length of 140 frames at 4 FPS. Subsequently, we estimate camera poses using Depth Anything 3 (DA3) [Lin et al., 2025] to extract continuous action labels. Detailed visualizations concerning action extraction are deferred in Sec. A.3. Finally, language instructions are manually annotated by human experts. This pipeline establishes the largest real-world VLN dataset to date. We will release this dataset to benefit the community.

3.3 Stage 1: T2V $\rightarrow$ I2V

To achieve a balance between computational overhead and video generation fidelity, we adopt Wan2.1-1.3B T2V [Wan et al., 2025] model as our backbone. Wan utilizes a 3D causal VAE [Kingma and Welling, 2014] structure to compress the spatio-temporal dimensions. Specifically, given an input video $V \in \mathbb{R}^{(1+T)\times H\times W\times 3}$, Wan-VAE encodes it into latent chunks $C \in \mathbb{R}^{1+T/4, H/8, W/8, 16}$, resulting in $(1 + T/4)$ chunks where the shape of each chunk is $[H/8, W/8, 16]$. By default, all subsequent training processes are performed at the chunk level.

Note that the T2V model is designed to generate the future mainly based on language instructions rather than visual inputs. Hence, the first stage involves adapting the backbone from T2V to image-to-video (I2V) to ensure the consistency of the generated future with the initial observation.

We retain the original flow matching [Lipman et al., 2023] objective of Wan for fine-tuning. During training, given a chunk latent at an arbitrary timestep c_T, the following sparse chunk latents $x_1 = [c_{T+1}, c_{T+2}, c_{T+5}, c_{T+8}, ..., c_{T+20}]$, a random noise $x_0 \sim \mathcal{N}(0, I)$, and a timestep $t \in [0, 1]$ sampled from a logit-normal distribution, an intermediate latent x_t is obtained as the training input. x_t is defined as a linear interpolation between x_0 and x_1:

$$x_t = tx_1 + (1 - t)x_0. \tag{1}$$

The ground-truth velocity v_t is defined as:

$$v_t = \frac{dx_t}{dt} = x_1 - x_0. \tag{2}$$

The loss function is formulated as mean square error between the model output and the velocity v_t:

$$\mathcal{L}_{\text{stage1}} = \mathbb{E}_{x_0, x_1, l, c_T, t}||u(x_t, l, c_T, t; \theta) - v_t||^2, \tag{3}$$

where l is the language embedding of umT5 [Chung et al., 2023], θ is the model weights, and $u(x_t, l, c_T, t; \theta)$ denotes the predicted velocity.

3.4 Stage 2: History Injection

A critical distinction between navigation foundation models and previous vision-language-action models lies in the necessity of incorporating the entire history of observations [Kim et al., 2024, Bu et al., 2025b, Jiang et al., 2026, Wang et al., 2026]. Unlike LLMs that can directly absorb a long sequence of image tokens, VGMs lack this capability to process this input [Wan et al., 2025, Wu et al., 2025]. We draw inspiration from CDiT [Bar et al., 2025] architecture to inject the history information with efficient training and inference computational overhead. Specifically, we introduce an additional

cross-attention block within each transformer block of Wan backbone to explicitly inject history information. To preserve the generative priors of the fine-tuned I2V model, we initialize the final linear layer of these newly added cross-attention blocks with zeros.

Nevertheless, the whole history input presents another challenge due to its excessive length and high dimensionality. To efficiently extract spatio-temporal features, we employ a two-step strategy that utilizes a Q-Former [Li et al., 2023] to process features along the temporal dimension and subsequently applies a Video-Former [Hu et al., 2025] for features along the spatial dimension.

We denote the history embedding processed by Q-Former and Video-Former at an arbitrary timestep as h_T, the training objective is formulated as:

$$\mathcal{L}_{\text{stage2}} = \mathbb{E}_{x_0, x_1, l, c_T, h_T, t} ||u(x_t, l, c_T, h_T, t; \theta) - v_t||^2. \tag{4}$$

3.5 Stage 3: Diffusion Distillation

Unlike manipulation tasks, which involve limited visual changes [Wu et al., 2024, Kim et al., 2024, Shi et al., 2025] and allow high-fidelity reconstruction with few denoising steps [Hu et al., 2025, Bu et al., 2025a], navigation tasks involve highly dynamic scene transitions [Zhang et al., 2025a, 2026b]. Consequently, generating high-fidelity future in navigation scenarios via few-step denoising is inherently difficult, posing significant challenges for real-world deployment. While video generation-based methods have been explored in autonomous driving [Gao et al., 2024, Zhao et al., 2025], they still suffer from high inference latency, requiring tens of seconds to minutes for generation.

To address this efficiency bottleneck, we adapt PCM [Wang et al., 2024] to flow-matching paradigm for distilling history-injected I2V model. In this setup, we designate the history-injected I2V model as the teacher model and initialize an architecturally identical student model with the same weights. We then partition the noise schedule into 4 phases, where the student model learns to predict the solution point of each phase on the probability flow ODE [Song et al., 2021b, 2023] trajectory of the teacher model [Wang et al., 2024]. By minimizing the consistency loss between adjacent timesteps, we progressively distill the inference steps from $N = 50$ to $M = 4$.

3.6 Stage 4: Action Learning

Drawing inspiration from VPP [Hu et al., 2025], we freeze the distilled I2V model and employ an inverse dynamics paradigm to predict continuous actions. Specifically, the generated sparse future and the language instruction are injected into the DiT-based action head via cross-attention. Nevertheless, we observe a clear visual discrepancy between the generated and the original ground-truth future frames, which leads to a misalignment between the synthesized dynamics and the original action labels. To combat this inconsistency, we employ DA3 to relabel the generated future frames, thereby ensuring the action supervision is precisely aligned.

We adopt DDIM [Song et al., 2021a] to reconstruct the relabeled actions $\overline{a}_0$ from noised action $\overline{a}_k = \sqrt{\overline{\beta}_k \overline{a}_0} + \sqrt{1 - \overline{\beta}_k}\epsilon$, where ϵ denotes white noise and $\overline{\beta}_k$ is the noisy coefficient at timestep k. The training process can be formulated as learning a denoised D_ψ:

$$\mathcal{L}_{action}(\psi; A) = \mathbb{E}_{\overline{a}_0, \epsilon, k} ||D_\psi(\overline{a}_k, l, \overline{V}) - \overline{a}_0||^2, \tag{5}$$

where $\overline{V}$ denotes the generated sparse latent future observation.

4 Evaluations

To validate the efficacy of SparseVideoNav, we design our experiments to investigate: **1)** Can SparseVideoNav, as a framework grounded in video generation, exhibit foundational competency in following language instructions? (See Sec. 4.2); **2)** Does the sparse foresight successfully translate to superior performance in challenging beyond-the-view scenarios? (See Sec. 4.2 and Sec. 4.3); **3)** Does our architectural design strikes a good balance between efficiency and performance? (See Sec. 4.3); **4)** Can SparseVideoNav demonstrate clear scalability with increased data, adaptability to dynamic obstacles, and robustness to camera height variations? (See Sec. 4.3 and Sec. A.6)

Table 1: **Quantitative results of zero-shot performance on diverse real-world scenes.** SparseVideoNav achieves unanimously SOTA performance across all the scenes on both instruction-following navigation (IFN) and beyond-the-view navigation (BVN) tasks. The highest success rate across SparseVideoNav and all baselines is highlighted in **bold**. (Numbers represent **success rate** in %)

	Indoors				Outdoors				Night				**Average**	
	Room		Lab Bldg		Yard		Park		Square		Mountain		Total	
Method	IFN	BVN	IFN	BVN	IFN	BVN	IFN	BVN	IFN	BVN	IFN	BVN	IFN	BVN
Baselines														
Uni-NaVid [Zhang et al., 2025a]	20	5	10	0	10	0	20	10	0	0	0	0	10.0	2.5
StreamVLN [Wei et al., 2026b]	45	10	40	15	30	5	50	30	20	0	**25**	0	35.0	10.0
InternVLA-N1 [Team, 2025]	25	5	15	0	35	5	30	**40**	0	0	0	0	17.5	8.3
SparseVideoNav	**55**	**25**	**55**	**30**	**50**	**20**	**65**	**40**	**50**	**20**	**25**	**15**	**50.0**	**25.0**
Ablation Study														
(a) *w distilled 4 steps cont. 2*	25	0	20	0	10	0	35	15	5	0	0	0	15.8	2.5
(b) *w distilled 4 steps cont. 10*	50	10	40	5	35	5	45	40	30	5	20	5	36.7	11.7
(c) *w/o distilled 50 steps cont. 20*	70	35	65	45	55	30	75	60	75	20	35	25	62.5	35.8
(d) *SparseVideoNav w/o Former*	40	15	60	25	40	20	60	45	45	20	25	10	45.0	22.5

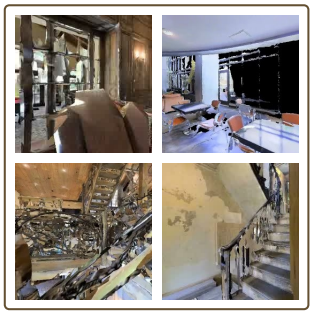

Figure 4: **Scenes of Habitat.**

Table 2: **Quantitative results of all methods on R2R-CE.**

Method	R2R Val-Unseen			
	NE↓	OS↑	SR↑	SPL↑
Uni-NaVid [Zhang et al., 2025a]	5.58	53.3	47.0	42.7
InternVLA-N1 [Team, 2025]	4.72	60.4	52.0	45.6
StreamVLN [Wei et al., 2026b]	4.89	60.6	54.4	52.1
SparseVideoNav	4.17	56.1	48.6	47.4

4.1 Experimental Setup

Baselines. To demonstrate the superiority of the VGM backbone, we choose three SOTA LLM-based models for comparison: **1)**Uni-Navid [Zhang et al., 2025a] uses video-based LLM to unify diverse VLN tasks; **2)**StreamVLN [Wei et al., 2026b] is designed to accelerate the inference through KV cache. **3)**InternVLA-N1 [Team, 2025] uses the video generation objective to co-train the backbone.

Evaluation Protocols. To assess the navigation capabilities in real-world environments, we select six diverse unseen scenes across three categories: **Indoors** (Room, Lab Building), **Outdoors** (Yard, Park), and **Night** (Square, Mountain) to test the **zero-shot** generalization capability. For each scene, we design four distinct navigation tasks, consisting of two standard instruction-following navigation (IFN) tasks and two challenging beyond-the-view navigation (BVN) tasks. Detailed specifications for all the tasks are deferred in Sec. A.2. To ensure statistical reliability, each model is tested 10 times per task, resulting in a total of **240 trials** per model. We report the **Success Rate** averaged across task types for each scene. We observe that LLM-based baselines often stop with a lateral orientation towards the target, whereas our model typically stops facing the target directly. To ensure a fair comparison, we define success solely based on proximity, where a trial is considered successful if the agent stops within 1.5 meters of the target. To minimize the influence of time and weather, we conduct comparative tests for all models within the same time window to ensure consistent environmental conditions. Hardware deployment details are deferred in Sec. A.5.

To assess foundational language understanding capabilities in the established benchmark, we leverage the VLN-CE R2R [Krantz and Lee, 2022]. We only use R2R, RxR [Ku et al., 2020b], R2R-EnvDrop-CE [Tan et al., 2019], and ScaleVLN [Cruz et al., 2023] to train our model from scratch without any dagger data.

4.2 Main Results

As shown in Table 1, SparseVideoNav achieves unanimously SOTA zero-shot performance across all real-world scenes on both IFN and BVN tasks. In high-fidelity real-world scenes, the true potential of VGM can be fully unleashed. Compared to the strongest baseline StreamVLN, SparseVideoNav

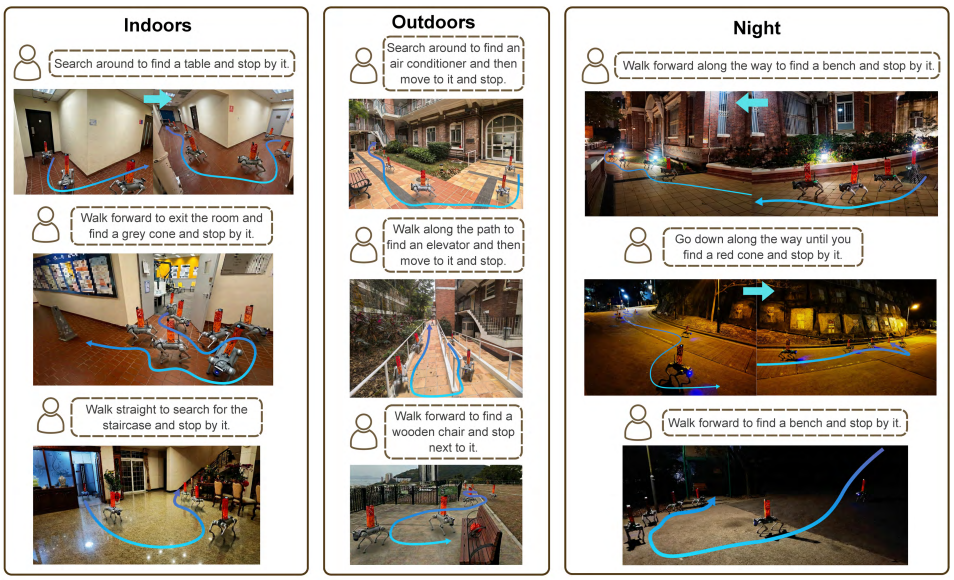

Figure 5: **Qualitative results of zero-shot beyond-the-view navigation in challenging, unstructured environments.** SparseVideoNav successfully navigates through challenging scenarios, including dead ends, narrow accessible ramp, and hillside with high inclination angles.

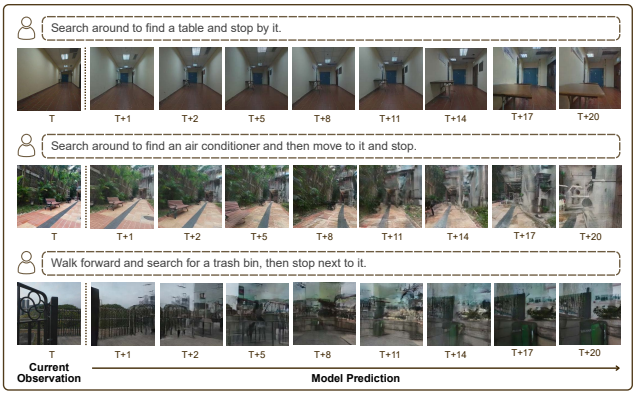

Figure 6: **Analysis of video generation results** during zero-shot deployment in beyond-the-view navigation.



Figure 7: **Ablation study on diffusion distillation (Stage 3).**

achieves a significant performance improvement, with the average success rate by **+15.0%** on IFN and **+15.0%** on challenging BVN. Notably, the diminished visibility in challenging night environments exacerbates the short-sighted limitation, precipitating a systematic failure across all established baselines on BVN tasks. In contrast, by leveraging robust long-horizon guidance, SparseVideoNav stands as the sole method capable of navigating to these distant goals in such extreme environments. Furthermore, as shown in Sec. A.6, even when baselines are fine-tuned on our self-collected dataset, they still show no significant performance improvement, demonstrating that the superiority of SparseVideoNav stems from its well-designed architecture.

As shown in Table 2, SparseVideoNav achieves reasonable performance on VLN-CE R2R even without any dagger data. We observe that the poor rendering quality of Habitat [Savva et al., 2019] introduces a severe visual gap, as shown in Fig. 4. Unlike LLM-based methods that map observations into abstract **semantic** spaces, VGMs are intrinsically more sensitive to visual fidelity as they rely on **pixel-level reconstruction**. Since Wan is pre-trained on high-fidelity videos [Wan et al., 2025], forcing it to reconstruct low-quality scenes would constrain its pre-trained priors. Nevertheless, SparseVideoNav exhibits the smallest discrepancy between success rate and success weighted by path length among all baselines. This minimal gap underscores the profound advantage brought by the long-horizon foresight when guiding the action generation.

Qualitative Results. To further showcase the superiority of SparseVideoNav in BVN, we visualize several trajectories in Fig. 5. SparseVideoNav successfully navigates through these challenging scenarios, including dead ends, narrow accessible ramp, and hillside with high inclination angles.

Analysis. To better understand why SparseVideoNav excels in BVN tasks, we present model prediction results (See Fig. 6) during deployment and several representative failure cases of prior methods (See Fig. 1). As LLM-based baselines rely on short-horizon supervision, they suffer from inherent short-sighted limitations, leading to unexpected turning under long-range uncertainty and premature trapping in dead ends. Conversely, by integrating this guidance with closed-loop feedback, SparseVideoNav can effectively mitigate this limitation. To further inspire future explorations, we detail failure cases of SparseVideoNav in Sec. A.1.

4.3 Ablation Study

Diffusion Distillation. As evidenced by Fig. 7, distillation enables VGM to match the visual fidelity of the original 50-step inference with only 4 denoising steps. Crucially, as illustrated in Fig. 8(a), the inclusion of distillation yields a substantial computational advantage, accelerating inference by approximately $10\times$ with slight performance compromise (See *Ablation Study* of Table 1 variant (c)).

Sparse Video Generation. To demonstrate the advantage of our proposed sparse design, we devise the following variants for comparative analysis:

a) *w distilled 4 steps cont. 2* denotes distilled 4-step variant for generating 2 continuous chunks. Variant (a) is specifically designed to emulate the effects of short-sighted limitations typically observed in LLM baselines.

b) *w distilled 4 steps cont. 10* denotes distilled 4-step variant for generating 10 continuous chunks.

c) *w/o distilled 50 steps cont. 20* denotes undistilled 50-step variant for generating 20 continuous chunks, where variant (c) can be seen as an oracle compared to SparseVideoNav.

d) *SparseVideoNav w/o Former* denotes our method without Q-Former and Video-Former.

As shown in the *Ablation Study* of Table 1, constrained by the same short-horizon supervision issue as LLM baselines, variant (a) yields suboptimal performance. While extending the horizon in variant (b) offers partial mitigation, a distinct performance discrepancy remains compared to SparseVideoNav. In further comparison with variant (c), SparseVideoNav demonstrates substantial efficiency gains, specifically a $1.7\times$ increase in inference speed (See Fig. 8(b)) and a $1.4\times$ reduction in the cumulative convergence time across training stages 1 and 2 (See Fig. 1). These results show a favorable efficiency-performance trade-off with minimal compromise.

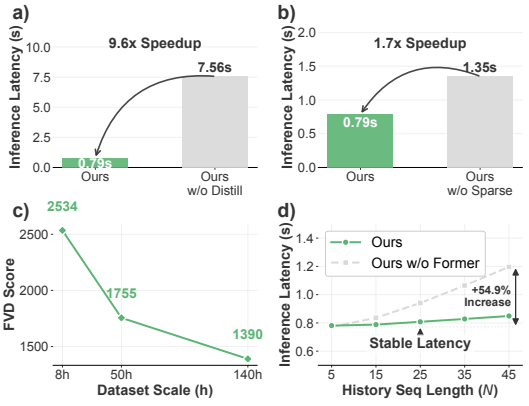

Figure 8: **Ablation study** on a) data scalability and computational overhead comparison over b) sparse design, c) distillation, and d) history compression.

Data Scalability. We conduct the whole training process except Stage 4 on varying data scales: 8h, 50h, and 140h, with additional 3h of unseen data as a validation set to compute FVD [Unterthiner et al., 2019]. As shown in Fig. 8(c), the consistent downward trend of the FVD curve highlights the scalability of SparseVideoNav, demonstrating the capability to absorb large-scale real-world navigation data.

History Compression Strategy. Fig. 8(d) presents the inference latency across varying history lengths. Former structure decouples latency from history length, ensuring stable inference latency. In contrast to the variant without Former, which incurs a $+54.9\%$ latency penalty at $N = 45$, SparseVideoNav achieves superior efficiency and slightly better performance (See *Ablation Study* of Table 1 variant (d)).

Pretraining from T2V to I2V. To validate the necessity of Stage 1, we benchmark the training time required for convergence under two variants: **a)** our proposed progressive adaptation; **b)** training

Stage 2 directly from scratch. Experiments are conducted on a cluster of 32 NVIDIA H200 GPUs. The results show that variant (b) requires 64 hours to reach convergence, whereas our progressive adaptation strategy converges in 32 hours. This yields a $2\times$ training speedup with similar performance, demonstrating that Stage 1 serves as an efficient initialization to significantly reduce the overall computational overhead.

5 Conclusion and Limitation

We introduce SparseVideoNav, pioneering the use of VGM as the primary backbone for action generation in VLN. By shifting to long-horizon sparse foresight, it overcomes short-sightedness and reduces computational overhead, significantly outperforming SOTA LLM baselines in zero-shot real-world evaluations. Despite its promise, limitations exist: **1)** Our 140-hour dataset is not web-scale [Bar et al., 2025, Mei et al., 2026], requiring future scaling. **2)** Inference latency slightly trails LLM paradigms [Wei et al., 2026b], necessitating accelerated distillation and quantization for VGM.

Acknowledgment

This study is supported by National Natural Science Foundation of China (62206172). This work is in part supported by the JC STEM Lab of Autonomous Intelligent Systems funded by The Hong Kong Jockey Club Charities Trust. We are grateful to Jiazhi Yang and Di Zhang for their valuable discussions, Haoguang Mai, Longyang Wu, Hongchen Li, Shuo Diao, Haolin Ou, Ronghao Li, and members from OpenDriveLab for their assistance throughout this project. We also acknowledges DJI for funding support during the project.

References

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation World Models. In *CVPR*, 2025. 3, 5, 10, 15

Hongzhe Bi, Hengkai Tan, Shenghao Xie, Zeyuan Wang, Shuhe Huang, Haitian Liu, Ruowen Zhao, Yao Feng, Chendong Xiang, Yinze Rong, et al. Motus: A Unified Latent Action World Model. In *CVPR*, 2026. 3

Qingwen Bu, Jia Zeng, Li Chen, Yanchao Yang, Guyue Zhou, Junchi Yan, Ping Luo, Heming Cui, Yi Ma, and Hongyang Li. Closed-Loop Visuomotor Control with Generative Expectation for Robotic Manipulation. In *NeurIPS*, 2024. 3

Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Xindong He, Xu Huang, et al. AgiBot World Colosseo: A Large-scale Manipulation Platform for Scalable and Intelligent Embodied Systems. In *IROS*, 2025a. 6

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. UniVLA: Learning to Act Anywhere with Task-centric Latent Actions. In *RSS*, 2025b. 5

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 *ICCV*, 2025. 3

Li Chen, Chonghao Sima, Kashyap Chitta, Antonio Loquercio, Ping Luo, Yi Ma, and Hongyang Li. Intelligent Robot Manipulation Requires Self-Directed Learning. *Authorea Preprints*, 2025. 3

An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Xueyan Zou, Jan Kautz, Erdem Biyik, Hongxu Yin, Sifei Liu, and Xiaolong Wang. NaVILA: Legged Robot Vision-Language-Action Model for Navigation. In *RSS*, 2025. 2

Hyung Won Chung, Xavier Garcia, Adam Roberts, Yi Tay, Orhan Firat, Sharan Narang, and Noah Constant. UniMax: Fairer and More Effective Language Sampling for Large-Scale Multilingual Pretraining. In *ICLR*, 2023. 5

Michael Cruz, Christopher Lee, Saurabh Aggarwal, and Robert Martin. Scaling Autoregressive Multi-modal Models: Pretraining and Instruction Tuning. *arXiv preprint arXiv:2309.02591*, 2023. 7

Ruiyuan Gao, Kai Chen, Bo Xiao, Lanqing Hong, Zhenguo Li, and Qiang Xu. MagicDriveDiT: High-resolution Long Video Generation for Autonomous Driving with Adaptive Control. *arXiv preprint arXiv:2411.13807*, 2024. 3, 6

Zeying Gong, Rong Li, Tianshuai Hu, Ronghe Qiu, Lingdong Kong, Lingfeng Zhang, Yiyi Ding, Leying Zhang, and Junwei Liang. Stairway to Success: Zero-Shot Floor-Aware Object-Goal Navigation via LLM-Driven Coarse-to-Fine Exploration. *arXiv preprint arXiv:2505.23019*, 2025. 3

Noriaki Hirose, Amir Sadeghian, Marynel Vázquez, Patrick Goebel, and Silvio Savarese. GoNet: A Semi-Supervised Deep Learning Approach for Traversability Estimation. In *IROS*, 2018. 5

Noriaki Hirose, Dhruv Shah, Ajay Sridhar, and Sergey Levine. Sacson: Scalable Autonomous Control for Social Navigation. In *IEEE RA-L*, 2023. 5

Yucheng Hu, Yanjiang Guo, Pengchao Wang, Xiaoyu Chen, Yen-Jen Wang, Jianke Zhang, Koushil Sreenath, Chaochao Lu, and Jianyu Chen. Video Prediction Policy: A Generalist Robot Policy with Predictive Visual Representations. In *ICML*, 2025. 2, 3, 4, 6, 18

Xijie Huang, Weiqi Gai, Tianyue Wu, Congyu Wang, Zhiyang Liu, Xin Zhou, Yuze Wu, and Fei Gao. NavDreamer: Video Models as Zero-Shot 3D Navigators. In *RAL*, 2026. 3

Haoran Jiang, Jin Chen, Qingwen Bu, Li Chen, Modi Shi, Yanjie Zhang, Delong Li, Chuanzhe Suo, Chuang Wang, Zhihui Peng, and Hongyang Li. WholeBodyVLA: Towards Unified Latent VLA for Whole-Body Loco-Manipulation Control. In *ICLR*, 2026. 5

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An Open-Source Vision-Language-Action Model. In *CoRL*, 2024. 5, 6

Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. In *ICLR*, 2014. 5

Jacob Krantz and Stefan Lee. Sim-2-Sim Transfer for Vision-and-Language Navigation in Continuous Environments. In *ECCV*, 2022. 7

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-Across-Room: Multilingual Vision-and-Language Navigation with Dense Spatiotemporal Grounding. In *EMNLP*, 2020a. 2

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-Across-Room: Multilingual Vision-and-Language Navigation with Dense Spatiotemporal Grounding. In *EMNLP*, 2020b. 7

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models. In *ICML*, 2023. 4, 6, 18

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, et al. Causal World Modeling for Robot Control. *arXiv preprint arXiv:2601.21998*, 2026. 3

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the Visual Space from Any Views. *arXiv preprint arXiv:2511.10647*, 2025. 5, 16, 19

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow Matching for Generative Modeling. In *ICLR*, 2023. 5

Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. InstructNav: Zero-shot System for Generic Instruction Navigation in Unexplored Environment. In *CoRL*, 2024. 3

Ilya Loshchilov and Frank Hutter. Decoupled Weight Decay Regularization. In *ICLR*, 2019. 17

Yanghong Mei, Yirong Yang, Longteng Guo, Qunbo Wang, Ming-Ming Yu, Xingjian He, Wenjun Wu, and Jing Liu. UrbanNav: Learning Language-Guided Urban Navigation from Web-Scale Human Trajectories. In *AAAI*, 2026. 10

William Peebles and Saining Xie. Scalable Diffusion Models with Transformers. In *ICCV*, 2023. 18

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A Platform for Embodied AI Research. In *ICCV*, 2019. 5, 8

Dhruv Shah, Benjamin Eysenbach, Nicholas Rhinehart, and Sergey Levine. Rapid Exploration for Open-World Navigation with Latent Goal Models. In *CoRL*, 2021. 5

Modi Shi, Li Chen, Jin Chen, Yuxiang Lu, Chiming Liu, Guanghui Ren, Ping Luo, Di Huang, Maoqing Yao, and Hongyang Li. Is Diversity All You Need for Scalable Robotic Manipulation? *arXiv preprint arXiv:2507.06219*, 2025. 6

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising Diffusion Implicit Models. In *ICLR*, 2021a. 6

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *ICLR*, 2021b. 6

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency Models. In *ICML*, 2023. 6

Akash Srivastava, Lazar Valkov, Chris Russell, Michael U Gutmann, and Charles Sutton. VEEGAN: Reducing Mode Collapse in Gans Using Implicit Variational Learning. In *NeurIPS*, 2017. 5

Hao Tan, Licheng Yu, and Mohit Bansal. Learning to Navigate Unseen Environments: Back Translation with Environmental Dropout. In *NAACL*, 2019. 7

InternVLA-N1 Team. InternVLA-N1: An Open Dual-System Navigation Foundation Model with Learned Latent Plans, 2025. URL https://internrobotics.github.io/internvla-n1.github.io/static/pdfs/InternVLA_N1.pdf. 2, 3, 7, 15, 18

Hoang Thanh-Tung and Truyen Tran. Catastrophic Forgetting and Mode Collapse in GANs. In *IJCNN*, 2020. 5

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. FVD: A New Metric for Video Generation, 2019. URL https://openreview.net/forum?id=rylgEULtdN. 9

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and Advanced Large-Scale Video Generative Models. *arXiv preprint arXiv:2503.20314*, 2025. 2, 5, 8, 17, 21

Fu-Yun Wang, Zhaoyang Huang, Alexander Bergman, Dazhong Shen, Peng Gao, Michael Lingelbach, Keqiang Sun, Weikang Bian, Guanglu Song, Yu Liu, et al. Phased Consistency Models. In *NeurIPS*, 2024. 6

Shaoan Wang, Jiazhao Zhang, Minghan Li, Jiahang Liu, Anqi Li, Kui Wu, Fangwei Zhong, Junzhi Yu, Zhizheng Zhang, and He Wang. TrackVLA: Embodied Visual Tracking in the Wild. *arXiv preprint arXiv:2505.23189*, 2025. 2

Ziye Wang, Modi Shi, Chaojun Ni, Jiazhi Yang, Mengdi Li, Zhizhong Su, Tianwei Lin, and Hongyang Li. NativeMEM: Native Memory Compression for Long-Horizon Robotic Manipulation. In *CoRL*, 2026. 5

Meng Wei, Chenyang Wan, Jiaqi Peng, Xiqian Yu, Yuqiang Yang, Delin Feng, Wenzhe Cai, Chenming Zhu, Tai Wang, Jiangmiao Pang, et al. Ground Slow, Move Fast: A Dual-System Foundation Model for Generalizable Vision-and-Language Navigation. In *ICLR*, 2026a. 2

Meng Wei, Chenyang Wan, Xiqian Yu, Tai Wang, Yuqiang Yang, Xiaohan Mao, Chenming Zhu, Wenzhe Cai, Hanqing Wang, Yilun Chen, et al. StreamVLN: Streaming Vision-and-Language Navigation via Slowfast Context Modeling. In *ICRA*, 2026b. 2, 3, 7, 10

Bing Wu, Chang Zou, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Jack Peng, Jianbing Wu, Jiangfeng Xiong, Jie Jiang, et al. Hunyuanvideo 1.5 Technical Report. *arXiv preprint arXiv:2511.18870*, 2025. 2, 5

Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing Large-Scale Video Generative Pre-training for Visual Robot Manipulation. In *ICLR*, 2024. 6

Jiazhi Yang, Kashyap Chitta, Shenyuan Gao, Long Chen, Yuqian Shao, Xiaosong Jia, Hongyang Li, Andreas Geiger, Xiangyu Yue, and Li Chen. ReSim: Reliable World Simulation for Autonomous Driving. In *NeurIPS*, 2025. 3

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World Action Models are Zero-shot Policies. *arxiv preprint arXiv:2602.15922*, 2026. 3

Naoki Yokoyama, Sehoon Ha, Dhruv Batra, Jiuguang Wang, and Bernadette Bucher. VLFM: Vision-Language Frontier Maps for Zero-Shot Semantic Navigation. In *ICRA*, 2024. 3

Bangguo Yu, Yuzhen Liu, Lei Han, Hamidreza Kasaei, Tingguang Li, and Ming Cao. VLN-Game: Vision-Language Equilibrium Search for Zero-Shot Semantic Navigation. *arXiv preprint arXiv:2411.11609*, 2024. 3

Hai Zhang, Siqi Liang, Yanjun Guo, Shoufeng Zhang, Yunhao Wang, Hangyu Du, Haoguang Mai, Hongchen Li, Jiazhi Yang, Jialong Zeng, Yu Hu, Chen Chen, Gangcheng Jiang, Zishun Zhuo, Lirui Zhao, Zhigang Li, Yu Qiao, Zhenjie Yang, Fu Zhang, and Hongyang Li. Compass 1: Towards Open-World and Persistent Embodied Navigation from Monocular RGB Videos. *arXiv preprint arXiv:2611.12087*, 2026a. 2

Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-NaVid: A Video-based Vision-Language-Action Model for Unifying Embodied Navigation Tasks. In *RSS*, 2025a. 2, 3, 5, 6, 7

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. In *ICLR*, 2026b. 2, 5, 6

L Zhang, X Hao, Q Xu, Q Zhang, X Zhang, P Wang, J Zhang, Z Wang, S Zhang, and R Xu. A Novel Memory Representation via Annotated Semantic Maps for VLM-Based Vision-and-Language Navigation. In *ACL*, 2025b. 3

Guosheng Zhao, Xiaofeng Wang, Zheng Zhu, Xinze Chen, Guan Huang, Xiaoyi Bao, and Xingang Wang. DriveDreamer-2: LLM-Enhanced World Models for Diverse Driving Video Generation. In *AAAI*, 2025. 3, 6

A Appendix

In the Appendix, we first compile "motivating" questions in Sec. A.1. Task specifications details in Sec. A.2 are presented to supplement Sec. 4.1 in the main paper. We then provide additional elaborations on data curation details, implementation details and hardware deployment details in Sec. A.3, Sec. A.4 and Sec. A.5 respectively. Next, we discuss the adaptability, robustness of SparseVideoNav and other experiments in Sec. A.6. The Appendix concludes with Sec. A.7, which details the ethics statement and licenses.

A.1 Motivating Questions

Q1: *What are the detailed failure cases of SparseVideoNav?*

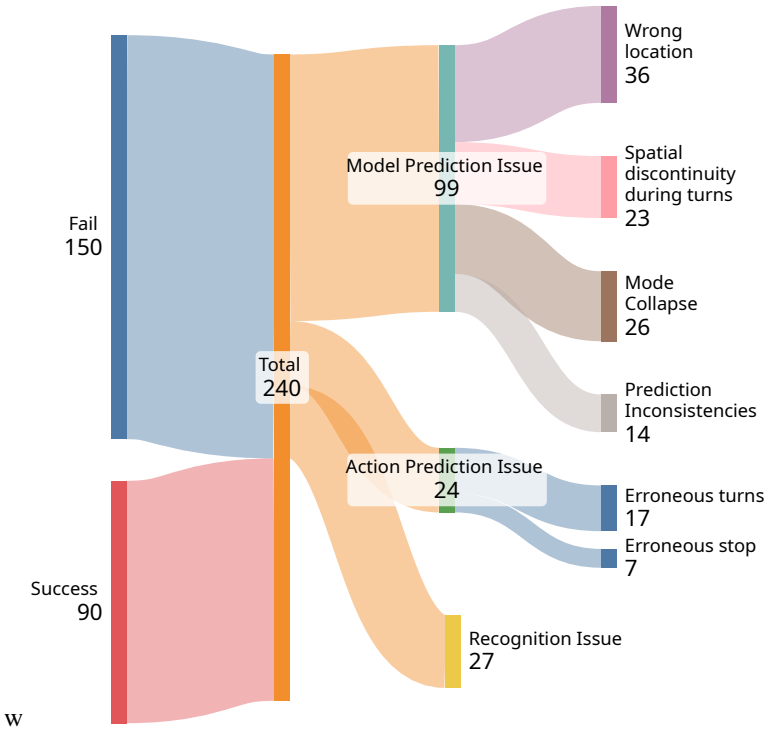

Figure A-1: **Detailed analysis of the failure cases of SparseVideoNav.**

We conduct a comprehensive statistical analysis of the failure cases observed across all 240 real-world evaluation trajectories. As illustrated in Fig. A-1, out of 150 failure instances, we categorize them into three primary groups:

- **Model Prediction Issues (99 cases):** This category encompasses failures stemming from the video generation process. The most prevalent error is wrong location (36 cases). This typically occurs in expansive, open environments where the target position "imagined" by the generative model misaligns with the real world (*e.g.,* early turning at a certain crossing). The agent ultimately fails to localize the true destination within the context limits. The second factor is mode collapse (26 cases, occurring more frequently in night scenes), which we attribute to the currently limited training data volume (approximately 140 hours). Additionally, spatial discontinuity during turns (23 cases) occurs due to the constraints of the fixed sparse generation interval. Finally, prediction inconsistencies (14 cases) arise when the navigation intent predicted at a certain timestep misaligns with subsequent predictions during the BVN task. These observations strongly motivate our future efforts in exploraing lifelong navigation, scaling the dataset volume and introducing adaptive sparsity mechanisms to better anchor predictions in complex open worlds.

- **Action Prediction Issues (24 cases):** This group involves misalignments between the predicted visual future and the generated actions. Specific manifestations include erroneous turns (17 cases)—such as visually predicting a right turn but outputting a left-turn command—and erroneous stops (7 cases), where a visually predicted slight turn prematurely triggers a stop action. We hypothesize that this stems from our decoupled design choice (further discussed in Q2). While decoupling ensures critical training stability for the video generation model, relying on a separately trained lightweight action head may limit its generalization capabilities. Future work exploring an effective vision-action joint training architecture in embodied navigation could potentially alleviate this gap.
- **Recognition Issues (27 cases):** In some instances, VGM fails to identify the target object category, even when situated in close proximity. We attribute this representational shortcoming to the relatively small parameter size (1.3B) of our chosen base model. Future directions to resolve this include leveraging larger-scale base models coupled with advanced quantization and distillation algorithms to maintain inference efficiency.

Q2: *What is the structural necessity of the four-stage training pipeline?*

Decoupling the video model training (Stage 2) and action training (Stage 4) is **a pragmatic design choice prioritizing training stability**. We initially considered vision-action joint optimization. However, unlike robotic manipulation where the camera view remains largely static and even 1-step denoising can extract sufficient features, real-world navigation involves drastic scene variations. In such dynamic settings, even performing 10+ denoising steps struggles to extract usable representations, making simultaneous optimization for both high-fidelity video generation and precise action prediction highly susceptible to instability. To ensure robust learning of the foundational video representations, we opt for this decoupled paradigm as an effective first step.

As for adaptation (Stage 1) and distillation (Stage 3), they are served for computational efficiency (See Sec. 4.3), which do not introduce great complexity.

Q3: *What is the distinction between Navigation World Model [Bar et al., 2025] and SparseVideoNav?*

To the best of our knowledge, SparseVideoNav is the first to apply a video generation model as the primary backbone for action generation in VLN. The distinction between SparseVideoNav and prior Navigation World Models stems from a fundamental modality mismatch:

- **Navigation World Model**: It is strictly **action-conditioned**. The architecture relies on low-level action inputs to predict the immediate next frame, rather than generating long-horizon videos.
- **SparseVideoNav**: Conversely, our framework is **language-conditioned**. It directly processes high-level language intents to achieve long-horizon video foresight, completely eliminating the need for action inputs.

Furthermore, we demonstrate that utilizing longer sequence supervision during training yields notable performance improvements (See Sec. 4.3). This effectively validates the significance of sparse video generation in effectively extending the prediction span.

Q4: *What about the object and language diversity of our collected real-world dataset?*

Regarding object diversity, we aim for open-world navigation. The targets extend far beyond "benches or chairs" in common simulation benchmarks, encompassing diverse **authentic physical entities** (e.g., fire hydrants, vehicles, boats, fence, elevators, road signs, bus station). While their locations may appear "predictable", they strictly follow natural real-world semantic priors.

Regarding language diversity, we explicitly **depart from** the previous paradigm of dense instructions. Our motivation is to minimize the human interaction burden by using simple and high-level intents.

Q5: *Compared to the pixel-goal guidance in InternVLA-N1 [Team, 2025], what is the advantage of sparse video guidance?*

The point-goal generated by language models are fundamentally confined to the current view. Consequently, it often fails to assist the agent in escaping when trapped in dead ends. In contrast, video

generation models can leverage long-horizon video guidance to imagine a backtracking trajectory to exit dead ends, thereby mitigating this perspective limitation and leading to more robust solutions for BVN tasks.

Q6: *What is the complexity of our real-world BVN tasks?*

Table A-I: Task settings and environmental characteristics.

Task setting	Shortest-path distance	Initial target visibility	Environment structure
Indoor BVN	8–15 m	Invisible in all tasks	Semi-open, with accessible branches
Outdoor(Night) BVN	15–35 m	Invisible in all tasks	Open or semi-open, with multiple accessible branches

As shown in Tab. A-I, these BVN tasks are conducted in open or semi-open environments containing multiple accessible branches. The agent receives no intermediate language guidance indicating which branch to take or when to turn. If it enters an incorrect branch, it must recognize the dead-end, return to the intersection, reorient itself, and continue in the correct direction. Such exploration and recovery can add approximately 10–20 m to the executed trajectory beyond the shortest path.

Moreover, the target is invisible in the initial observation for every BVN task. In more challenging cases, the target remains unobserved for over 10-20 seconds. For reference, even when moving continuously without turning, the robot travels at most approximately 10 m within 10 seconds. The agent must therefore preserve its navigation intent and select actions over an extended period without directly observing the target.

The 256 x 256 input resolution further limits the visual detail available during large turns and long-range observation. During a substantial turn, a traversable side branch may become apparent before the target or any target-relevant landmark becomes recognizable, encouraging the agent to explore an incorrect route. This difficulty is amplified at night, where low illumination and reduced contrast make distant targets and landmarks substantially harder to identify and further prolong the target-unobserved period.

Concerning the number of turns, this is policy-dependent rather than a fixed property of a BVN task because no intermediate language instruction specifies when or where the agent should turn. For example, several baselines become trapped in repeated turning near the starting position or make unexpected turns before reaching a relevant intersection.

These properties distinguish our BVN evaluation from standard IFN and many existing real-world demonstrations. To the best of our knowledge, real-world navigation tasks at this scale for the trained end-to-end model, combined with an initially invisible target, sparse high-level language, and multiple accessible branches, have rarely been evaluated systematically in existing literature. We will update the per-task distances and environment layouts in the revision.

A.2 Task Specifications and Details

Fig. A-2 and Fig. A-3 provide the complete specifications for all 24 zero-shot evaluation tasks across six real-world scenes. Each scene contains 2 instruction-following navigation (IFN) tasks and 2 beyond-the-view navigation (BVN) tasks.

A.3 Data Curation Details

As illustrated in Figure A-4, our pipeline efficiently transforms raw video streams into high-quality training image-action pairs. The process begins with temporal sampling, where raw videos are decimated to 4 FPS to reduce redundancy. We then utilize Depth Anything 3 (DA3) [Lin et al., 2025] to estimate the 6-DoF camera extrinsics for each frame, reconstructing the 3D trajectory of the agent. Finally, action extraction is performed by calculating the relative pose transformations between frames. These 3D motions are projected onto the local XY plane to derive continuous action labels $(\Delta x, \Delta y, \Delta\theta)$, while static segments and views with extreme pitch angles are filtered out to ensure high-quality selection.

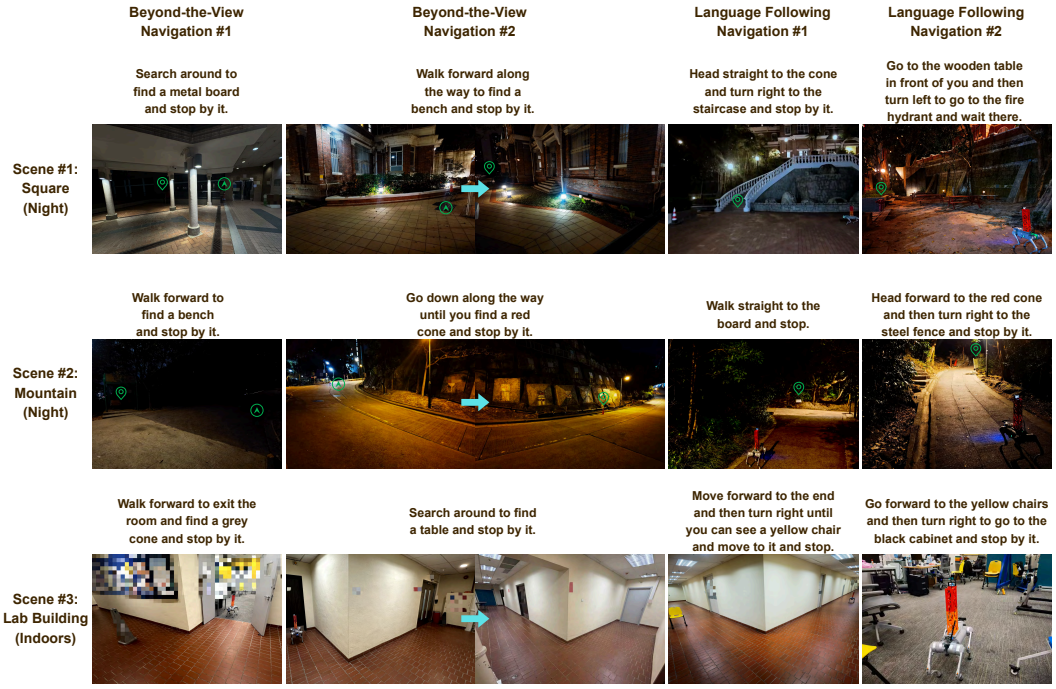

Figure A-2: **Task specifications.** We present the language instructions, the initial positions of robot dog and the task settings for each scene. When the precise locations can not be clearly seen, we utilize a green arrow icon to denote the initial positions and a location pin to mark the destination for visualization clarity.

Table A-II: **Hyperparameter configurations for the four-stage training pipeline.**

Stage	Batch Size	Learning Rate	LR Scheduler	Trainable Params	Sampling Strategy	Discrete Steps
Stage 1	32	1×10^{-5}	Constant	1.3B	FM (Uniform)	1000
Stage 2	32	1×10^{-5}	Constant	1.8B	FM (Uniform)	1000
Stage 3	32	5×10^{-6}	Constant	1.7B	PCM (Distill)	50
Stage 4	256	5×10^{-5}	Cosine	23.4M	DDIM	100

Table A-III: **Hyper-parameters of Q-Former and Video-Former used for history compression.**

Q-Former Configuration		Video-Former Configuration	
Hidden Dim	512	Hidden Dim	512
Layers	4	Layers	6
Heads	8	Heads	8
Num. Latents	10240	Num. Latents	2560

A.4 Implementation Details

Training Configurations. Our model is trained using the AdamW optimizer [Loshchilov and Hutter, 2019] on a cluster of 32 NVIDIA H200 GPUs, with the full four-stage pipeline requiring approximately 64 hours to complete. All training hyper-parameters of each stage is presented in Table A-II.

Network Architecture Configurations. Our system integrates three core components: a video generation backbone, a history compression module, and an action prediction head. For the video backbone (Stage 1), we directly adopt the Wan2.1 T2V-1.3B [Wan et al., 2025] architecture. To handle

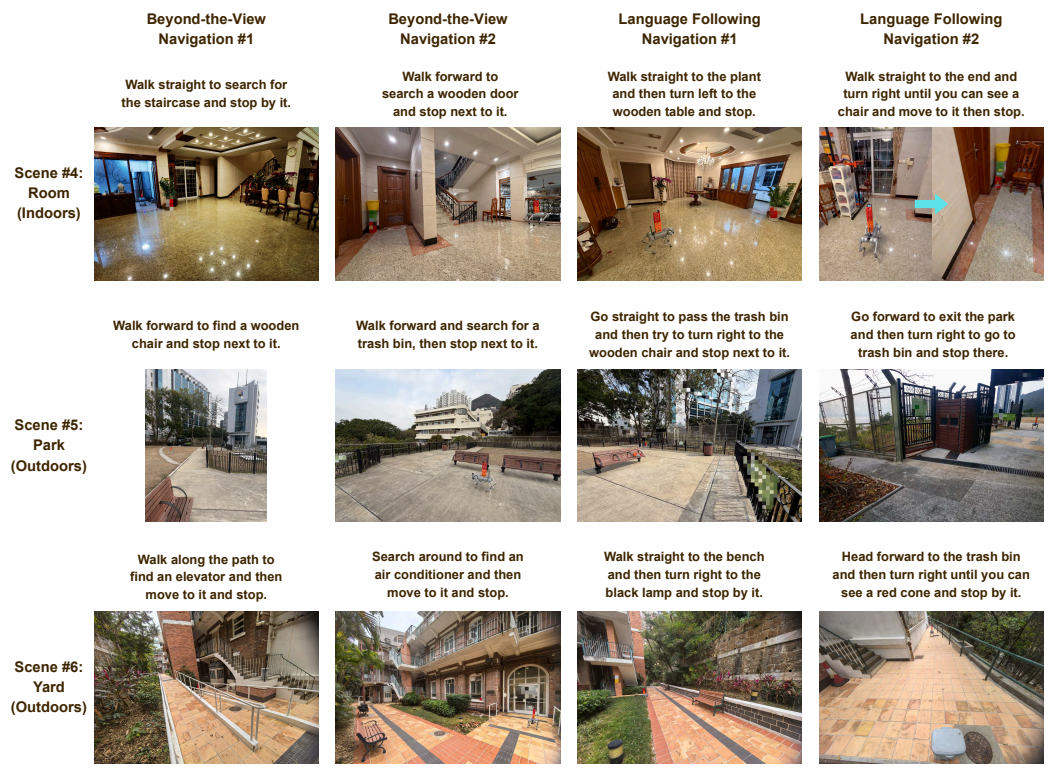

Figure A-3: **Task specifications.** We present the language instructions, the initial positions of robot dog and the task settings for each scene.

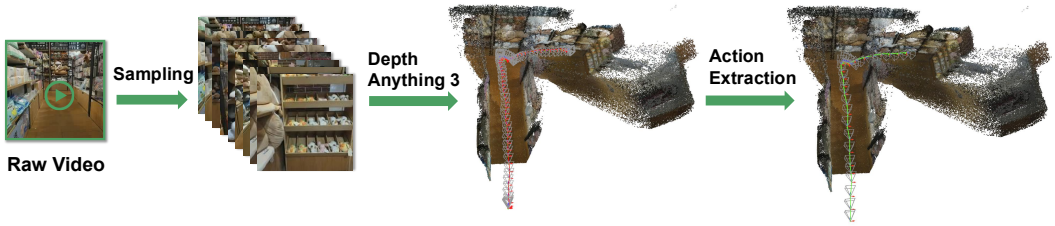

Figure A-4: **Data Curation Pipeline.** The workflow consists of three stages: (1) temporal sampling from raw video; (2) pose estimation via Depth Anything 3; and (3) extrinsic-based action extraction.

high-dimensional historical context, we implement a two-stage compression strategy: a Q-Former [Li et al., 2023] first reduces temporal redundancy, followed by a Video-Former [Hu et al., 2025] that performs $4\times$ spatial downsampling. This reduces the token burden to 2,560 while preserving essential context. We present the detailed hyperparameters of Q-Former and Video-Former in Table A-III.

For navigation control, we design an inverse dynamics-based action head that couples a feature-aggregating Video-Former with a Diffusion Transformer (DiT) [Peebles and Xie, 2023]. Video-Former encodes spatiotemporal features from 640 sparse future latent tokens. These features then serve as the cross-attention context for the DiT to predict continuous 8-step action trajectories. Detailed parameters for these modules are summarized in Table A-IV.

A.5 Hardware Deployment Details

We perform real-world experiments based on Unitree Go2 robotic dog. The robot is equipped with an upward facing camera DJI Osmo Action 4 for stabilized RGB observations. However, given that InternVLA-N1 [Team, 2025] requires depth input, we adhere to its original configuration by employing an Intel® RealSense™ D455 camera, mounted with a 15° downward pitch to acquire

Table A-IV: **Hyper-parameters of the inverse dynamics-based action head**, including the Action Video-Former and the Diffusion Transformer.

Action Video-Former		**Diffusion Transformer**	
Hidden Dim	256	Hidden Dim	256
Layers	8	Layers	12
Heads	8	Heads	8
Num. Latents	640	Num. Actions	8



Figure A-5: **Discussion of SparseVideoNav under dynamic pedestrian interference.** SparseVideoNav successfully avoids the oncoming pedestrian and reaches the door.

RGB-D observations. The camera is secured to the back of Go2 using a custom 3D-printed bracket, positioned at a height of approximately 1m above the ground. This mounting height is standardized across all evaluated methods. We deploy SparseVideoNav and all the baselines on a remote workstation with an RTX 4090 GPU. The Go2 robot continuously sends visual observations to the server. Upon receiving the images, the server processes them and generates corresponding navigation action commands, which are then sent back to Go2 for execution.

A.6 Additional Experiments

Controlled Real-World-Data Comparisons. To directly examine whether gains of SparseVideoNav primarily result from access to our 140-hour real-world dataset, we additionally fine-tuned StreamVLN, and InternVLA-N1 using the same real-world data. All entries below denote success rate (%). "Real" denotes training or fine-tuning with our real-world navigation data.

Table A-V: Performance comparison across different scenarios against baselines finetuned by our collected real-world data.

Method	**Lab IFN**	**Lab BVN**	**Yard IFN**	**Yard BVN**	**Mtn. IFN**	**Mtn. BVN**
SparseVideoNav	**55**	**30**	**50**	**20**	**25**	**15**
StreamVLN (Real)	40	10	**50**	0	20	0
InternVLA-N1 (Real)	25	5	40	0	5	0

Tab. A-V shows the performance comparsion on baselines finetuned by our collected data. In BVN scenarios, we frequently observe unstable or repeated turning when the target remains outside the current view for all VLM-based baselines, consistent with the failure mode analyzed in the main paper. By contrast, SparseVideoNav behaves more consistently across IFN and BVN settings, including when the competing methods are given access to our real-world data. This controlled data-access comparison indicates that the observed improvement can be attributed to our improved model architecture.

Dynamic Pedestrians Avoidance.

Given that DA3 [Lin et al., 2025] struggles with reliable action estimation in the presence of frontal dynamic pedestrians, we filter out such trajectories to preserve the accuracy of the actions. Remarkably, despite this exclusion, SparseVideoNav exhibits an emergent capability to dynamically

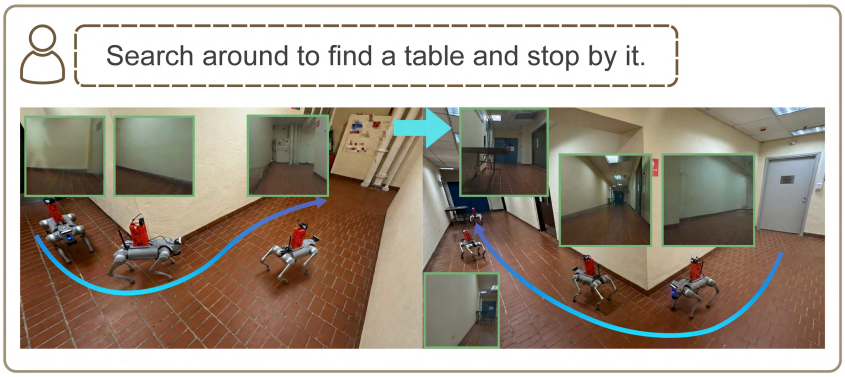

Figure A-6: **Discussion of SparseVideoNav with the camera fixed at 50cm.** Images in green boxes demonstrate the predictions generated by SparseVideoNav. SparseVideoNav exhibits strong robustness against camera height variations.

Table A-VI: Pedestrian avoidance performance across different environments.

Method	Lab pedestrian avoidance	Yard pedestrian avoidance	Mountain pedestrian avoidance
SparseVideoNav	7/10	7/10	5/10

avoid pedestrians during deployment, underscoring its strong adaptability and generalization, as shown in Fig. A-5 and Tab. A-VI.

Camera Height Insensitivity. We further observe that while LLM-based paradigm is notably susceptible to camera height shifts, video generation-based paradigm maintains robust against these discrepancies. Despite being trained on data collected at an approximate height of 1m, SparseVideoNav successfully performs navigation with a fixed camera height of 50cm, as shown in Fig. A-6 and Tab. A-VII.

Longer Cross-Room and Multi-Turn Evaluation Tests. We further conducted an additional evaluation on three-stage IFN setting. As shown in A-VIII, SparseVideoNav succeeds in 4/10 trials, compared with 1/10 for CorrectNav, 3/10 for OmniNav, and 2/10 for StreamVLN. It shows that SparseVideoNav remains effective on longer trajectories requiring multiple sequential turns.

Sparse Interval. We further conducted an additional evaluation on sparse interval. As shown in Tab. A-IX, when interval equals to 1, it is beneficial for IFN, but the short supervision coverage reduces the performance of BVN. As the interval gradually increases, artifacts easily arise during the continuous movement, leading to a performance drop for both IFN and BVN. Thus, interval equals to 3 is a favorable choice.

Denoising Steps. As shown in Tab. A-X, reducing the number of steps to $M = 2$ improves inference speed but degrades the generated future representations and downstream navigation performance. Increasing it to $M = 8$ improves generation quality only marginally while increasing inference latency. The results therefore indicate that $M = 4$ provides a suitable balance between navigation performance and real-time inference efficiency.

Action Head. Our motivation for using a diffusion-based action head follows prior findings in manipulation: diffusion policies can feature multimodal action distributions, whereas deterministic regression tends to average multiple feasible action modes.

This property is relevant to navigation because the same high-level intent and predicted future can admit multiple valid local trajectories, particularly near open intersections or during obstacle

Table A-VII: Performance evaluation under different camera heights.

Camera height	Lab IFN	Lab BVN	Yard IFN	Yard BVN	Mountain IFN	Mountain BVN
SparseVideoNav (1.0 m)	55	30	50	20	25	15
SparseVideoNav (0.7 m)	50	20	55	20	15	5

Table A-VIII: Success Rate on longer cross-room and multi-turn evaluation tasks, with *prompt: Walk exit the room to the yellow chair and then turn right to go to the black box, and then turn left to enter the lab to get to the metal cart and stop.*

Method	SR
SparseVideoNav	**4/10**
CorrectNav(Real)	1/10
OmniNav(Real)	3/10
StreamVLN(Real)	2/10

Table A-IX: Quantitative ablation of the sparse interval.

Sparse interval	Lab IFN	Lab BVN	Yard IFN	Yard BVN	Mountain IFN	Mountain BVN
1	**65**	10	**50**	10	**30**	0
3	55	**30**	**50**	**20**	25	**15**
5	30	5	25	5	5	0

Table A-X: Ablation on distillation steps.

Distillation steps	Lab IFN	Lab BVN	Yard IFN	Yard BVN	Mountain IFN	Mountain BVN	Inference latency
M = 2	10	0	5	5	0	0	**0.48 s**
M = 4	55	30	**50**	**20**	**25**	15	0.79 s
M = 8	**60**	**35**	**50**	**20**	20	**20**	1.41 s

Table A-XI: Performance comparison with different action heads across scenarios.

Action head	Lab IFN	Lab BVN	Yard IFN	Yard BVN	Mountain IFN	Mountain BVN
Diffusion	**55**	**30**	**50**	**20**	**25**	**15**
MLP regression	20	5	35	10	5	5

avoidance. We added a direct comparison against an MLP regression head while keeping the VGM backbone and training data unchanged.

As shown in Tab. A-XI, the diffusion-based action head consistently performs better than direct MLP regression.

A.7 Data Ethics and License

Privacy Protection and De-identification. To strictly protect individual privacy, we implement a robust and irreversible de-identification processing pipeline. All captured video frames are processed by human expert to identify all sensitive information. All sensitive information, including human faces and vehicle identifiers, is permanently blurred before being used for model training or stored in our servers. We ensure that no identifiable personal information is present in the final dataset.

Data Collection Ethics. Our data collection involves human operators recording in diverse public environments. All operators are briefed on ethical guidelines, ensuring that they follow local regulations and avoid recording in restricted or private areas without explicit authorization. Data collection is performed using a handheld DJI Osmo Action 4 in real-world environments. This manual approach ensures data diversity and procedural transparency throughout the recording process.

License. We build upon Wan2.1 [Wan et al., 2025], which is under Apache License 2.0. The code and dataset for SparseVideoNav would be open-sourced under CC BY-NC-SA 4.0 License.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, the main claims are clearly laid out at the end of introduction and validated in Sec. 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Yes, see the end of Sec. 5 and our analysis of failure cases in Sec. A.1.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This paper does not have any theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Yes, see our appendix for training details, implementation details, and hardware deployment details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Please see the link below the title.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: Yes, see our appendix for training details, implementation details, and hardware deployment details.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: A common practice in the field of embodied navigation is to evaluate models and directly report the performance.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Yes, see our experiments for the details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We make sure to preserve anonymity and conform to the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The positive impacts are discussed in Sec. 1.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Yes, we will open-source our dataset after processing it with de-identification pipeline.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes, see Sec. A.7.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Yes, we will open-source our dataset alongside a thorough usage document.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subject.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLM is used only for polishing purpose.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.