# SCALENAV: SCALABLE INTERNET VIDEO PRETRAINING FOR LANGUAGE-CONDITIONED LOCAL NAVIGATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

We study language-conditioned local navigation, where a robot must navigate to a text-specified target visible in its initial observation and stop at the described location. End-to-end navigation policies require diverse training data to generalize across scenes, yet real-robot data is costly to collect and simulated data leaves a sim-to-real gap. Internet egocentric videos offer a scalable source of real-world navigation experience. Our key insight is to learn goal-directed visual dynamics from Internet egocentric videos, and use these dynamics to support robot action generation. To this end, we propose ScaleNav, a unified world-action framework with three-stage training. We first perform instruction-conditioned video pretraining on these videos to learn how observations evolve as the camera approaches instructed targets across diverse real-world scenes. Action alignment then uses action-labeled simulated data to train the model to predict executable action chunks. Joint fine-tuning subsequently optimizes all trainable components together for navigation. ScaleNav achieves success rates of 86.5% on a single-view adaptation of Short-Horizon OVON and 94.9% on our HabitatGS-LN benchmark, outperforming the evaluated baselines on both benchmarks. We further deploy and evaluate ScaleNav on a real quadruped robot, with results supporting the model's ability to generalize to real-world navigation.

## 1 INTRODUCTION

Navigating to visible targets according to natural-language instructions is an important capability for robots. For example, a person may ask a robot to walk to the desk by the window and stop there. Similar instructions also arise in long-horizon systems, where a high-level planner, such as an agent based on a large language model (LLM), decomposes a task into a sequence of short-range goals involving visible objects or regions (Huang et al., 2023; Rajvanshi et al., 2024; Ding et al., 2025). We study this language-conditioned local navigation setting: the robot must navigate to a text-specified target visible in its initial field of view and stop at the described location.

One possible solution is to connect a grounding module to an existing navigation method (Shah et al., 2023a; Huang et al., 2023; Cai et al., 2024; Xu et al., 2025; Long et al., 2025; Cerdio et al., 2026), but errors in target grounding and spatial-goal conversion can lead to navigation errors. End-to-end policies provide an alternative by mapping visual observations and language instructions directly to actions (Zhang et al., 2024; Hirose et al., 2025; Zhang et al., 2025a; Wei et al., 2025; Zhang et al., 2026b; Zhu et al., 2026; Wang et al., 2026b; Huang et al., 2026). To generalize across real-world scenes, these policies require diverse training data (Shah et al., 2023b;c; Sridhar et al., 2024). However, real-robot action data is costly to collect across diverse environments (Liu et al., 2025; Cai et al., 2025), whereas simulated action data still leaves a sim-to-real gap (Cai et al., 2025; Xia et al., 2026).

Internet egocentric videos offer a scalable source of visual experience covering diverse real-world scenes and human navigation behavior. However, they lack corresponding robot action labels. Some methods attempt to leverage these videos by extracting intermediate motion-related supervision, such as estimated trajectories and geometric targets (Cheng et al., 2025; Liu et al., 2025; Hirose et al., 2025; Seneviratne et al., 2026; Zhou et al., 2026), or latent actions inferred from vi-



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

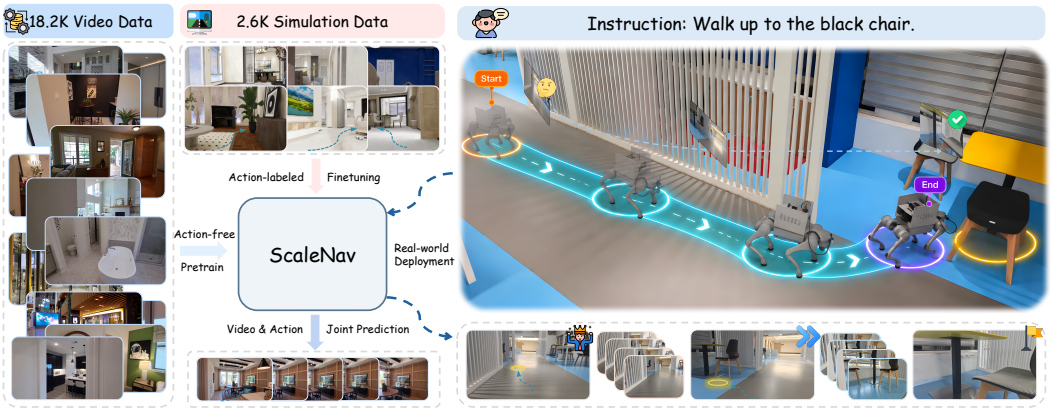



Figure 1: **Overview of ScaleNav.** ScaleNav learns navigation priors from Internet egocentric videos through action-free video pretraining, then uses action-labeled simulated data for action alignment and joint fine-tuning. It jointly predicts future observations and action chunks to follow navigation instructions. The right panel illustrates deployment on a real quadruped robot.

sual transitions (Bu et al., 2025). Errors or biases introduced during this extraction can propagate through the resulting supervision into subsequent policy learning.

To effectively learn navigation priors from these videos and translate them into executable actions, we propose ScaleNav, a framework for language-conditioned local navigation that combines a unified world-action model (WAM) with a three-stage training procedure. As illustrated in Figure 2, the model uses a shared Diffusion Transformer (DiT) (Peebles & Xie, 2023) to jointly predict future observations and action chunks. We first perform instruction-conditioned video pretraining on Internet egocentric videos to learn navigation priors from human navigation behavior in diverse real-world scenes. Action alignment then uses action-labeled simulated data to train the model to predict executable action chunks while keeping part of the model frozen. Finally, joint fine-tuning optimizes all trainable components together for video and action prediction in navigation. During navigation, the model predicts an action chunk from the instruction and visual observations, executes actions from the chunk, and repeats this process with updated observations until it reaches the target and stops.

For evaluation, we construct HabitatGS-LN, a benchmark for navigation to specific objects described in natural-language instructions. We also evaluate ScaleNav on a single-view adaptation of Short-Horizon OVON (Gong et al., 2026). ScaleNav achieves success rates (SR) of 86.5% on Short-Horizon OVON and 94.9% on HabitatGS-LN, outperforming the compared methods on both benchmarks. With identical action-labeled simulated data, incorporating Internet egocentric videos improves success by 5.3 and 4.5 percentage points on the two benchmarks, respectively. We further deploy and evaluate ScaleNav on a real quadruped robot in three indoor environments. ScaleNav achieves 83.3% overall SR and outperforms the evaluated baselines in all three environments.

Our contributions are as follows:

- We propose ScaleNav, a framework for language-conditioned local navigation that transfers navigation priors learned from diverse Internet egocentric videos through video pretraining into executable action chunks via action alignment.
- Experiments show that ScaleNav outperforms the compared methods in both navigation success rate and path efficiency on the two simulation benchmarks. Real-robot experiments further support ScaleNav's ability to generalize to real-world navigation.

## 2 Related Work

**Language-conditioned navigation.** Language-conditioned navigation connects language understanding with visual perception, planning, and action generation. Modular approaches combine pretrained foundation models with explicit maps, candidate waypoints, or planning compo-

nents (Shah et al., 2023a; Huang et al., 2023; Chen et al., 2024; Long et al., 2025; Cerdio et al., 2026). Learned vision-language-action policies adapt pretrained vision-language backbones to predict navigation actions from instructions and visual observations (Zhang et al., 2024; 2025a; Wei et al., 2025; Zhang et al., 2026b; Zhu et al., 2026; Wang et al., 2026b). Within this direction, dual-system architectures connect high-level semantic reasoning to low-level action generation (Wei et al., 2026; Gong et al., 2026). Within language-conditioned navigation, local navigation focuses on reaching a language-specified destination without a prescribed route. Methods for this setting learn language-conditioned goal-reaching policies (Hirose et al., 2025; Glossop et al., 2026; Mei et al., 2026) or address local target grounding under occlusion (Gao et al., 2026). We focus on this local navigation setting.

**World-action models.** Using video prediction for robot control requires linking visual predictions to executable actions (Zhao et al., 2026). One approach generates visual plans and recovers actions through inverse dynamics or geometric estimation (Du et al., 2023; Ko et al., 2024; Chen et al., 2026; Lee et al., 2026). Related approaches condition their action decoders on predictive visual features or latents (Hu et al., 2025; Zhang et al., 2026a). Another line of work jointly models future observations and actions within a unified architecture (Guo et al., 2024; Zhu et al., 2025; Cen et al., 2025; Ye et al., 2026; Dong et al., 2026; Azuma et al., 2026; Yang et al., 2026; Zhou et al., 2026; Huang et al., 2026; Li et al., 2026). Following DreamZero, we adopt this joint video-action architecture for navigation.

**Learning from Internet egocentric videos.** Prior work in robot manipulation learns from action-free human videos through video pretraining (Wu et al., 2024; Cheang et al., 2024) or video-only cross-embodiment transfer (Ye et al., 2026). In navigation, existing approaches leverage these videos mainly in two ways. One line of work uses Internet videos to train navigation policies through intermediate motion-related supervision or latent-action representations. Geometry-based methods derive trajectory or geometric prediction targets from estimated camera motion or reconstructed scenes (Liu et al., 2025; Cheng et al., 2025; Mei et al., 2026; Han et al., 2025; Seneviratne et al., 2026; Zhou et al., 2026). Other methods obtain action annotations or teacher supervision from learned models (Lin et al., 2023; Hirose et al., 2025). Related video-based methods infer latent actions from visual transitions for policy pretraining (Ye et al., 2025; Bu et al., 2025). Errors or biases in these derived targets and representations can propagate into subsequent policy learning.

Another line of work uses Internet videos to learn world models for navigation planning. NWM learns a navigation world model from unlabeled videos without proxy motion targets (Bar et al., 2025). Garrido et al. (2026) learn a latent-action world model from Internet videos and adapt it to navigation by mapping robot actions to latent actions. These models evaluate candidate trajectories determined by a separate search procedure or policy. Better predictions therefore do not necessarily yield better navigation decisions when suitable candidates are missing. Our framework learns navigation priors from Internet egocentric videos and directly predicts executable action chunks, without constructing proxy motion targets for these videos or relying on an external planner at inference.

# 3 METHOD

## OVERVIEW

We study language-conditioned local navigation: given a text instruction and a stream of RGB observations, a robot navigates to a target object or region visible in its initial field of view and stops at the instructed location. A single action $a_t = (\Delta x_t, \Delta y_t, \Delta\psi_t)$ represents the motion between two consecutive frames, where $(\Delta x_t, \Delta y_t)$ is the planar displacement in the robot's local coordinates at the preceding frame and $\Delta\psi_t$ is the change in yaw. A sequence of $T$ actions forms an action chunk, denoted by $\tau = (a_1, \ldots, a_T)$. The robot collects RGB observations while executing each predicted action chunk, then uses the updated observations to predict the next chunk.

We use a world-action model with a shared DiT backbone to jointly predict future observations and action chunks (Section 3.1). Through three-stage training (Section 3.2), the model first learns navigation priors from Internet egocentric videos and is then adapted for action prediction using action-labeled simulated data. Section 3.3 describes how we construct both datasets.

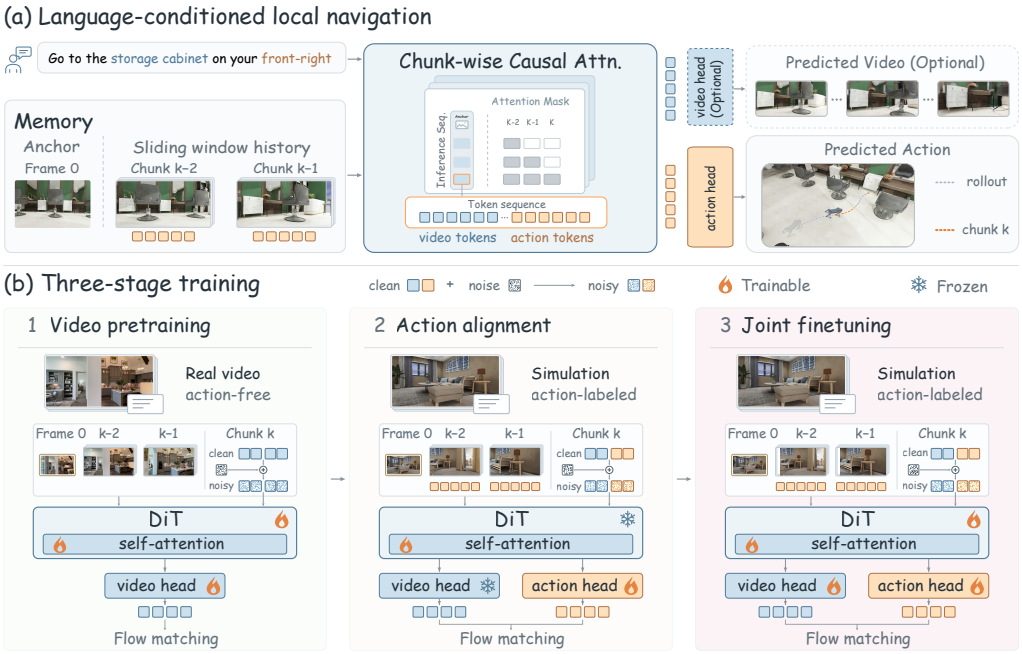



Figure 2: **Overview of ScaleNav. (a) Language-conditioned local navigation.** The model predicts video and action chunks from the instruction, current observation, and memory using chunk-wise causal attention. The memory contains an initial-frame anchor and a sliding window of recent video and action chunks. **(b) Three-stage training.** Video pretraining uses action-free real videos, followed by action alignment and joint fine-tuning with action-labeled simulation data. During action alignment, the self-attention layers within the DiT, the action head, and the action embedding MLP (not shown) are trainable. The snowflake on the outer DiT block marks the remaining DiT layers as frozen. Joint fine-tuning updates the DiT and both heads. For clarity, the initial bidirectional adaptation in Stage 1 and Internet egocentric video replay in Stages 2 and 3 are omitted.

### 3.1 Model Architecture

**Unified world-action model.** We follow DreamZero (Ye et al., 2026) and use a unified world-action model that jointly predicts future observations and actions with a shared DiT backbone. Because the two modalities share backbone parameters, action prediction can draw on visual priors learned from Internet egocentric videos. Specifically, we denote the clean variational autoencoder (VAE) (Kingma & Welling, 2013) latents of future observations by $z_0^v$ and the clean, normalized representation of action chunk $\tau$ by $z_0^a$. Their noisy counterparts at noise levels $t_v$ and $t_a$ are denoted by $z_{t_v}^v$ and $z_{t_a}^a$, respectively. The noisy video and action representations are embedded as tokens, concatenated, and jointly processed by the DiT, conditioned on the navigation instruction $l$, current observation $o$, and history. The video and action heads map the corresponding output tokens to flow velocities $v_\theta^v$ and $v_\theta^a$ for joint denoising, where $\theta$ denotes the model parameters. The denoised action representation is denormalized into an action chunk, and the video latents are optionally decoded by the VAE into future observations.

**Initial-frame anchor and recent history.** Task instructions are specified with respect to the initial observation, and the robot must use the visual context in that view to identify the intended navigation target. For example, in "walk over to the second door on your left", "second" refers to the ordering of doors in the initial view. To maintain accurate instruction interpretation as the viewpoint changes, we use an anchor mechanism that retains the initial frame's video tokens throughout navigation. We also maintain a sliding window containing multiple recent chunks, each consisting of observations and past actions. The initial-frame anchor and the recent chunks form the history condition $M$ to support coherent navigation across chunks. This history is passed to the DiT as noise-free tokens, which current-chunk tokens attend to through chunk-wise causal attention.

### 3.2 THREE-STAGE TRAINING

We use a three-stage training procedure to learn navigation priors from Internet egocentric videos and use them to support action chunk generation. Stage 1 pretrains the model to predict egocentric observations as the camera approaches and reaches the instructed target, conditioned on navigation instructions. Stage 2 introduces action-labeled simulated data for action alignment, learning action prediction with part of the model frozen. Stage 3 jointly fine-tunes the model for video and action prediction.

**Stage 1: Video pretraining.** Pretrained video models provide spatiotemporal priors for action generation (Ye et al., 2026). However, their text conditioning, training-data selection, and prediction schemes differ from the requirements of downstream navigation. First, their text conditions typically describe scene content and motion (Kong et al., 2024; Yang et al., 2025), whereas navigation instructions specify a target to reach. Second, during data preparation, camera-dominated footage with little subject motion and clips with pronounced camera shake are sometimes down-weighted or excluded (Wan et al., 2025; Yang et al., 2025), limiting exposure to viewpoint changes and camera shake, both of which are common during robot navigation. Finally, these models commonly generate entire clips bidirectionally (Yang et al., 2025; Yin et al., 2025), whereas navigation requires updating predictions as new observations arrive.

To bridge these differences and further learn navigation priors from human navigation in diverse real-world scenes, we continue pretraining Wan2.2-TI2V-5B (Wan et al., 2025) on our Internet egocentric videos paired with navigation instructions. To avoid changing the data distribution and prediction structure simultaneously, we first train on these video-instruction pairs using Wan's original bidirectional text-image-to-video (TI2V) formulation. Throughout Stage 1, we condition on the navigation instruction and current observation, keeping the observation latent noise-free. We then continue training with chunk-wise prediction, using the initial-frame anchor and recent video chunks as noise-free history, corresponding to the video component of $M$. Chunk-wise causal attention allows current-chunk tokens to attend to historical video tokens while retaining bidirectional attention within the current chunk.

Throughout video pretraining, we use a video flow-matching objective (Lipman et al., 2023) with no action tokens in the input. For the future video latents $z_0^v$, we sample a noise level $t_v \in (0, 1)$ and Gaussian noise $\epsilon^v \sim \mathcal{N}(0, I)$, and construct $z_{t_v}^v = (1 - t_v)z_0^v + t_v\epsilon^v$. The model predicts the corresponding flow velocity $v_\theta^v$ with the objective

$$\mathcal{L}_{\text{video}} = \mathbb{E}\left[\left\|v_\theta^v - (\epsilon^v - z_0^v)\right\|^2\right].$$

The VAE and text encoder remain frozen throughout all three training stages.

**Stage 2: Action alignment.** To translate the navigation priors learned during video pretraining into executable actions, we train the model to jointly predict future observations and action chunks using action-labeled simulated data. During this alignment stage, we only train the randomly initialized action embedding multilayer perceptron (MLP) and action head together with the shared self-attention layers.

We independently sample video and action noise levels $t_v, t_a \in (0, 1)$ (Zhu et al., 2025) and add noise to the video latents as in Stage 1. For the normalized action chunk $z_0^a$, we sample Gaussian noise $\epsilon^a \sim \mathcal{N}(0, I)$ and construct $z_{t_a}^a = (1 - t_a)z_0^a + t_a\epsilon^a$. The model jointly predicts the video and action flow velocities $v_\theta^v$ and $v_\theta^a$. For batches of action-labeled simulated data, we normalize the inverse-loss weights discussed by Ma et al. (2024) to sum to one, yielding the training objective

$$\begin{aligned}
\mathcal{L} &= (1 - \lambda)\mathcal{L}_{\text{video}} + \lambda\mathcal{L}_{\text{action}}, \\
\mathcal{L}_{\text{action}} &= \mathbb{E}\left[\left\|v_\theta^a - (\epsilon^a - z_0^a)\right\|^2\right], \\
\lambda &= \text{sg}\left(\frac{\mathcal{L}_{\text{video}}}{\mathcal{L}_{\text{video}} + \mathcal{L}_{\text{action}}}\right).
\end{aligned}$$

Here, $\lambda$ is computed from the two losses on the current mini-batch, and sg denotes the stop-gradient operator. In Stages 2 and 3, we also replay Internet egocentric videos from the Stage 1 dataset to help retain the learned video priors, using only the video loss for these samples.

In addition, stopping after arrival is an important navigation capability. We model this behavior implicitly through action prediction. Specifically, we pad incomplete final chunks to the fixed length by repeating the last video frame and appending zero-motion actions $(0, 0, 0)$ before action normalization. To avoid potentially disturbing the pretrained video priors with artificially repeated final frames, we exclude latent positions marked as padding from $\mathcal{L}_{\text{video}}$ by applying a temporally max-pooled frame-validity mask. The padded zero actions remain included in $\mathcal{L}_{\text{action}}$ to supervise zero-motion outputs after arrival at the target.

**Stage 3: Joint fine-tuning.** Action alignment trains the action modules while keeping much of the video-pretrained model frozen. To further adapt the shared representations to robot navigation together with the action modules, we jointly fine-tune the model. Specifically, we unfreeze the remaining DiT modules and the video head, using the same joint training objective as in Stage 2.

### 3.3 Data

We construct two datasets for training: action-free Internet egocentric videos and action-labeled simulated data with aligned video and robot actions.

**Internet egocentric videos.** We seek egocentric video clips that capture human navigation behavior across diverse real-world scenes. We use the scene annotations provided by SpatialVID-HQ (Wang et al., 2026a) to select approximately 60K indoor clips from YouTube. For each clip, we provide uniformly sampled keyframes to a vision-language model (VLM) to identify candidate endpoints where the camera has arrived near a describable object or region. For each candidate endpoint, we take the corresponding object or region as the navigation target and use the VLM to search backward for the earliest keyframe in which the target is visible. Using the selected start and end timestamps, we extract the continuous segment from the source clip. For each extracted segment, we use the VLM to generate a navigation instruction describing the target by its category, attributes, or spatial relations.

To ensure accurate annotations and plausible navigation behavior, we then review the candidate segments and their instructions using the VLM. After filtering, the Internet egocentric video dataset contains 18.2K navigation segments. Detailed data construction and review procedures are provided in Appendix C.

**Action-labeled simulated data.** To obtain navigation data with aligned video and actions, we generate episodes in Habitat-GS (Xia et al., 2026) using indoor scenes from InteriorGS (Miao et al., 2026). We select targets from the initial observations and use the VLM with scene semantic annotations to generate unambiguous navigation instructions. Using ground-truth navmeshes, we plan smooth, collision-free trajectories with an offline 2D adaptation of SCAN-Planner (Zheng et al., 2026) and collect aligned observations and action labels along these trajectories. We review the resulting samples using criteria similar to those for the Internet egocentric videos, with scene semantic annotations providing additional evidence for instruction accuracy. This yields approximately 2.6K action-labeled episodes from 650 scenes. Further construction details are provided in Appendix C.

For both datasets, the VLM paraphrases the instructions of accepted samples to increase language diversity while preserving their meaning.

## 4 Experiments

### 4.1 Simulation Experiments

**Benchmarks.** We first evaluate on Short-Horizon OVON (Gong et al., 2026), which repositions the starts of HM3D-OVON Val-Unseen episodes (Yokoyama et al., 2024) to the first pose where the target becomes visible. The benchmark was originally developed for a three-view setting. We adapt it to our single-view setting, yielding 524 episodes across 36 HM3D scenes (Ramakrishnan et al., 2021), each paired with a navigation instruction referring to the target object.

Short-Horizon OVON evaluates navigation to category-level goals, but does not explicitly assess whether an agent can identify the intended object from the attributes and spatial relations described

Table 1: Comparison on Short-Horizon OVON and HabitatGS-LN. "Internet videos" refers to our curated Internet egocentric videos. Best results are in **bold**. Dashes indicate model variants not profiled separately for latency. † We evaluate a single-view subset of Short-Horizon OVON under our evaluation protocol.

| Method | Short-Horizon OVON† | | HabitatGS-LN | | Amortized latency (ms) ↓ |
|---|---|---|---|---|---|
| | SR (%) ↑ | SPL (%) ↑ | SR (%) ↑ | SPL (%) ↑ | |
| LocateAnything + NavDP | 74.4 | 37.2 | 82.8 | 61.0 | 184 |
| LocateAnything + ApexNav | 73.3 | 64.1 | 75.8 | 66.8 | 511 |
| StreamVLN (zero-shot) | 61.1 | 51.2 | 48.4 | 40.0 | 87 |
| Uni-NaVid (zero-shot) | 11.6 | 9.9 | 8.9 | 8.2 | 99 |
| StreamVLN (fine-tuned) | 79.6 | 62.9 | 75.8 | 59.5 | – |
| Uni-NaVid (fine-tuned) | 80.2 | 68.1 | 82.2 | 71.6 | – |
| Ours (w/o Internet videos) | 81.1 | 65.5 | 90.4 | 80.8 | – |
| **Ours** | **86.5** | **71.3** | **94.9** | **86.1** | **76** |

in an instruction. To complement this evaluation, we construct HabitatGS-LN, a benchmark for language-conditioned local navigation in photorealistic Habitat-GS environments (Xia et al., 2026), with 157 episodes across 45 held-out InteriorGS scenes (Miao et al., 2026). Detailed adaptation and construction procedures are provided in Appendix D.

**Evaluation protocol.** We conduct closed-loop evaluation on Short-Horizon OVON in Habitat (Savva et al., 2019) and on HabitatGS-LN in Habitat-GS (Xia et al., 2026). Episodes terminate according to each method's stopping rule or after 200 execution steps, without using ground-truth target proximity to trigger stopping. Each executed action counts as one step, including each action within a predicted chunk. Success requires stopping within the budget and within 1.25 m of a valid target, measured by the minimum Euclidean distance between their footprints. Valid targets are annotated goal instances of the specified category on Short-Horizon OVON and the instruction-specified instance on HabitatGS-LN. We report success rate (SR), success weighted by path length (SPL) (Anderson et al., 2018), and amortized inference latency per executed action on RTX 4090 GPUs. Profiling and evaluation details are provided in Appendices B and D, respectively.

**Baselines.** We compare with NavDP (Cai et al., 2025), ApexNav (Zhang et al., 2025b), StreamVLN (Wei et al., 2025), and Uni-NaVid (Zhang et al., 2025a). To adapt NavDP and ApexNav to our language-conditioned task, we use LocateAnything (Wang et al., 2026c) to ground instruction-specified objects in RGB observations. For NavDP, we back-project the initial detection using simulator-provided ground-truth depth to obtain a fixed world-frame goal. We express this goal as a 2D vector in the robot's current local frame at each step. For ApexNav, LocateAnything replaces the detectors in its category-level query pipeline. Both methods receive ground-truth depth with maximum ranges of 5 m for NavDP and 4.99 m for ApexNav, following their original evaluation settings.

For StreamVLN and Uni-NaVid, we evaluate both released checkpoints and versions fine-tuned on our action-labeled simulated data. Fine-tuning uses no additional DAgger (Ross et al., 2011) rollouts or reinforcement learning. Adaptation and fine-tuning details are provided in Appendix D.

We also evaluate a variant without additional training on our Internet egocentric videos. This variant skips Stage 1 and starts directly from the pretrained Wan weights, retaining Stages 2 and 3 on action-labeled simulated data but omitting video replay.

**Comparative results.** As shown in Table 1, ScaleNav outperforms the evaluated baselines in SR and SPL on both benchmarks. Compared with our variant without Internet egocentric videos, SR improves by 5.3 and 4.5 percentage points on Short-Horizon OVON and HabitatGS-LN, respectively. These gains support the benefit of learning navigation priors from Internet egocentric videos. Our

framework also achieves a comparatively low amortized inference latency of 76 ms per executed action.

### 4.2 Ablation Studies

We conduct the following ablation studies on HabitatGS-LN.

Table 2: Internet egocentric video utilization and component ablations on HabitatGS-LN. Video pretraining and Alignment denote Stages 1 and 2, respectively. Pseudo-action denotes action labels estimated from these videos, Anchor denotes the initial-frame anchor, and Replay denotes Internet egocentric video replay during Stages 2 and 3. Best results are highlighted in **bold**.

| Variant | Video pretraining | Pseudo-action | Anchor | Alignment | Replay | SR (%) ↑ | SPL (%) ↑ |
|---|---|---|---|---|---|---|---|
| w/o Internet videos | – | – | ✓ | ✓ | – | 90.4 | 80.8 |
| Pseudo-action training | – | ✓ | ✓ | ✓ | – | 73.9 | 67.1 |
| w/o video pretraining | – | – | ✓ | ✓ | ✓ | 90.4 | 83.1 |
| w/o initial-frame anchor | ✓ | – | – | ✓ | ✓ | 89.8 | 83.1 |
| w/o action alignment | ✓ | – | ✓ | – | ✓ | 93.6 | 85.1 |
| w/o video replay | ✓ | – | ✓ | ✓ | – | 94.3 | 84.2 |
| **Full model** | ✓ | – | ✓ | ✓ | ✓ | **94.9** | **86.1** |

**Internet egocentric video utilization.** We compare different ways of learning from Internet egocentric videos in Table 2. We use Depth Anything 3 (Lin et al., 2026) to estimate camera poses and derive pseudo action labels. The resulting video-action pairs are mixed with action-labeled simulated data during Stages 2 and 3, with equal sampling probability per clip and no Stage 1 pretraining. This yields 73.9% SR and 67.1% SPL, below the variant without Internet egocentric videos. Pose estimation errors may contribute to this degradation by introducing noise into action supervision.

We also consider jointly training on Internet egocentric videos and action-labeled simulated data during Stages 2 and 3, without a separate video pretraining stage. The Internet videos remain action-free and contribute only the video loss. This approach achieves 90.4% SR and 83.1% SPL, compared with 94.9% and 86.1% for our full model. These results suggest that dedicated video pretraining is important for learning navigation priors from Internet egocentric videos to support subsequent action generation.

**Component ablations.** Table 2 shows that removing the initial-frame anchor while retaining recent history reduces SR from 94.9% to 89.8%, supporting the benefit of preserving the initial visual reference. Skipping Stage 2 or omitting Internet egocentric video replay during Stages 2 and 3 yields 93.6% and 94.3% SR, respectively, suggesting modest benefits from action alignment and video replay.

### 4.3 Real-Robot Deployment

We further evaluate ScaleNav on a Unitree Go2 in an exhibition hall, an open lounge area, and an office area. Each method is evaluated on four tasks per environment with five trials per task, totaling 60 trials. We compare ScaleNav with its variant without Internet egocentric videos, StreamVLN, and Uni-NaVid, using the same initial poses and instructions.

We manually assess success using the stopping and footprint-distance criteria in Section 4.1. Further experimental details are provided in Appendix E.

ScaleNav achieves 83.3% overall SR and the highest SR among the evaluated methods in all three environments (Table 3). These results support its ability to generalize to real-world navigation. Figure 3 shows qualitative results.

Table 3: Real-robot success rates (60 trials per method). Best results are in **bold**.

| Method | SR (%) ↑ | | | |
|---|---|---|---|---|
| | Hall | Lounge | Office | Overall |
| StreamVLN (fine-tuned) | 25.0 | 60.0 | 40.0 | 41.7 |
| Uni-NaVid (fine-tuned) | 60.0 | 75.0 | 20.0 | 51.7 |
| Ours (w/o Internet videos) | 75.0 | 60.0 | 60.0 | 65.0 |
| **Ours** | **85.0** | **85.0** | **80.0** | **83.3** |

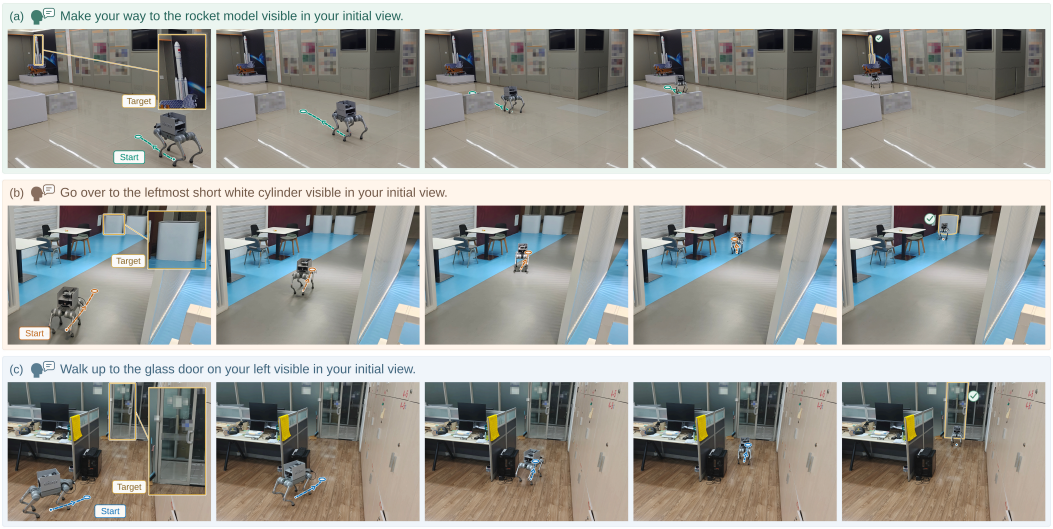



Figure 3: **Qualitative results of ScaleNav** in an exhibition hall (a), an open lounge area (b), and an office area (c). The robot follows language instructions to reach the specified targets.

## 5 Conclusion

We presented ScaleNav, a language-conditioned local navigation framework that learns navigation priors through Internet egocentric video pretraining and translates them into executable action chunks through action alignment and joint fine-tuning. Simulation experiments demonstrate the benefits of Internet egocentric videos and improvements over the evaluated baselines in success rate and path efficiency. Real-robot experiments further support generalization to real-world navigation. For limitations, our quantitative evaluation focuses on indoor navigation to initially visible object targets. Generalization to outdoor environments and navigation to region-level goals remain to be evaluated.

### AI use statement

In preparing this manuscript, we used generative AI tools to improve readability and organize the presentation. We also used a vision-language model to generate navigation instructions and assist with data screening, as described in Section 3.3. The authors reviewed and verified the AI-assisted manuscript text and take responsibility for the final submission.

## References

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018.

Daichi Azuma, Taiki Miyanishi, Koya Sakamoto, Shuhei Kurita, Yaonan Zhu, Petr Khrapchenkov, Motoaki Kawanabe, Yusuke Iwasawa, and Yutaka Matsuo. NavWAM: A navigation world action model for goal-conditioned visual navigation. *arXiv preprint arXiv:2606.13494*, 2026.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15791–15801. IEEE, 2025.

Qingwen Bu, Yanting Yang, Jisong Cai, Shenyuan Gao, Guanghui Ren, Maoqing Yao, Ping Luo, and Hongyang Li. Learning to Act Anywhere with Task-centric Latent Actions. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.014.

Wenzhe Cai, Siyuan Huang, Guangran Cheng, Yuxing Long, Peng Gao, Changyin Sun, and Hao Dong. Bridging zero-shot object navigation and foundation models through pixel-guided navigation skill. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5228–5234. IEEE, 2024.

Wenzhe Cai, Jiaqi Peng, Yuqiang Yang, Yujian Zhang, Meng Wei, Hanqing Wang, Yilun Chen, Tai Wang, and Jiangmiao Pang. NavDP: Learning sim-to-real navigation diffusion policy with privileged information guidance. *arXiv preprint arXiv:2505.08712*, 2025.

Jun Cen, Chaohui Yu, Hangjie Yuan, Yuming Jiang, Siteng Huang, Jiayan Guo, Xin Li, Yibing Song, Hao Luo, Fan Wang, et al. WorldVLA: Towards autoregressive action world model. *arXiv preprint arXiv:2506.21539*, 2025.

Esteban Padilla Cerdio, Boyang Sun, Marc Pollefeys, and Hermann Blum. OpenFrontier: General Navigation with Visual-Language Grounded Frontiers. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026. doi: 10.15607/RSS.2026.XXII.067.

Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, et al. GR-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024.

Jiaqi Chen, Bingqian Lin, Ran Xu, Zhenhua Chai, Xiaodan Liang, and Kwan-Yee Wong. MapGPT: Map-guided prompting with adaptive path planning for vision-and-language navigation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9796–9810, 2024.

Jie Chen, Yuxin Cai, Yizhuo Wang, Ruofei Bai, Yuhong Cao, Jun Li, Yau Wei Yun, and Guillaume Sartoretti. ImagiNav: Scalable embodied navigation via generative visual prediction and inverse dynamics. *arXiv preprint arXiv:2603.13833*, 2026.

An-Chieh Cheng, Yandong Ji, Zhaojing Yang, Zaitian Gongye, Xueyan Zou, Jan Kautz, Erdem Biyik, Hongxu Yin, Sifei Liu, and Xiaolong Wang. NaVILA: Legged Robot Vision-Language-Action Model for Navigation. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.018.

Hongyu Ding, Ziming Xu, Yudong Fang, You Wu, Zixuan Chen, Jieqi Shi, Jing Huo, Yifan Zhang, and Yang Gao. LaViRA: Language-vision-robot actions translation for zero-shot vision language navigation in continuous environments. *arXiv preprint arXiv:2510.19655*, 2025.

Yifei Dong, Fengyi Wu, Guangyu Chen, Lingdong Kong, Qiyu Hu, Yuxuan Zhou, Xu Zhu, Jingdong Sun, Jun-Yan He, Qi Dai, Alexander G. Hauptmann, and Zhi-Qi Cheng. Towards unified world models for visual navigation via memory-augmented planning and foresight, 2026. URL `https://arxiv.org/abs/2510.08713`.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. *Advances in Neural Information Processing Systems*, 36:9156–9172, 2023.

Christer Ericson. *Real-time collision detection*. CRC Press, 2004.

Xinyu Gao, Gang Chen, and Javier Alonso-Mora. BEACON: Language-conditioned navigation affordance prediction under occlusion. *arXiv preprint arXiv:2603.09961*, 2026.

Quentin Garrido, Tushar Nagarajan, Basile Terver, Nicolas Ballas, Yann LeCun, and Michael Rabbat. Learning latent action world models in the wild. In *Forty-third International Conference on Machine Learning*, 2026.

Catherine Glossop, William Chen, Arjun Bhorkar, Dhruv Shah, and Sergey Levine. CAST: Counterfactual labels improve instruction following in vision-language-action models. *IEEE Robotics and Automation Letters*, 2026.

Ruiyan Gong, Yingnan Guo, Junjun Hu, Jintao Kong, Xiaoxu Leng, Tianlun Li, Weize Li, Fei Liu, Zhicheng Liu, Jia Lu, et al. ABot-N1: Toward a general visual language navigation foundation model. *arXiv preprint arXiv:2607.10383*, 2026.

Yanjiang Guo, Yucheng Hu, Jianke Zhang, Yen-Jen Wang, Xiaoyu Chen, Chaochao Lu, and Jianyu Chen. Prediction with action: Visual policy learning via joint denoising process. *Advances in Neural Information Processing Systems*, 37:112386–112410, 2024.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025.

Mingfei Han, Liang Ma, Kamila Zhumakhanova, Ekaterina Radionova, Jingyi Zhang, Xiaojun Chang, Xiaodan Liang, and Ivan Laptev. RoomTour3D: Geometry-aware video-instruction tuning for embodied navigation. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 27586–27596. IEEE, 2025.

Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

Noriaki Hirose, Catherine Glossop, Ajay Sridhar, Oier Mees, and Sergey Levine. LeLaN: Learning a language-conditioned navigation policy from in-the-wild video. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 666–688. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/hirose25b.html.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

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 Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24328–24346. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/hu25g.html.

Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10608–10615. IEEE, 2023.

Yuehao Huang, Yunzi Wu, Xiaotao Zhang, Xinhai Li, Jiankun Dong, Jiajun Lv, Chi Zhang, Chenjia Bai, Yong Liu, and Xuelong Li. WNM-3D: A world navigation model with 3D scene conditioning for closed-loop VLN. *arXiv preprint arXiv:2608.07267*, 2026.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Po-Chen Ko, Jiayuan Mao, Yilun Du, Shao-Hua Sun, and Joshua B Tenenbaum. Learning to act from actionless videos through dense correspondences. In *International Conference on Learning Representations*, volume 2024, pp. 40938–40958, 2024.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. HunyuanVideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Hojin Lee, Sizhe Lester Li, Maximilian Hilger, Susie Lu, Achim J. Lilienthal, Vincent Sitzmann, and Daniel A. Duecker. Visual cue guided video planning for generalizable robot navigation, 2026. URL https://arxiv.org/abs/2609.16737.

Jingyu Li, Zhe Liu, Dongnan Hu, Junjie Wu, Zipei Ma, Wenxiao Wu, Chao Han, Zhihui Hao, Zhikang Liu, Kun Zhan, Jiankang Deng, Xiatian Zhu, and Li Zhang. Metis: A generalizable and efficient world-action model for autonomous driving and urban navigation, 2026. URL https://arxiv.org/abs/2606.15869.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Yang Zhao, Sida Peng, Hengkai Guo, Xiaowei Zhou, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the visual space from any views. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 141261–141285, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/e4cd50120b6d7e8daff1749d6bbaa889-Paper-Conference.pdf.

Kunyang Lin, Peihao Chen, Diwei Huang, Thomas H Li, Mingkui Tan, and Chuang Gan. Learning vision-and-language navigation from YouTube videos. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8283–8292. IEEE, 2023.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pp. 38–55. Springer, 2024.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6875–6885. IEEE, 2025.

Yuxing Long, Wenzhe Cai, Hongcheng Wang, Guanqi Zhan, and Hao Dong. InstructNav: Zero-shot system for generic instruction navigation in unexplored environment. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2049–2060. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/long25b.html.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Haoyu Ma, Jialong Wu, Ningya Feng, Chenjun Xiao, Dong Li, Jianye Hao, Jianmin Wang, and Mingsheng Long. HarmonyDream: Task harmonization inside world models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 33983–34007. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/ma24o.html.

Yanghong Mei, Yirong Yang, Longteng Guo, Qunbo Wang, Ming-Ming Yu, Xingjian He, Wenjun Wu, and Jing Liu. UrbanNav: Learning language-guided embodied urban navigation from web-scale human trajectories. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 18505–18513, 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. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 22945–22965, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/274d0146144643ee2459a602123c60ff-Paper-Conference.pdf.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4172–4182. IEEE, 2023.

Abhinav Rajvanshi, Karan Sikka, Xiao Lin, Bhoram Lee, Han-Pang Chiu, and Alvaro Velasquez. SayNav: Grounding large language models for dynamic planning to navigation in new environments. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 34, pp. 464–474, 2024.

Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-Matterport 3D Dataset (HM3D): 1000 large-scale 3D environments for embodied AI. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/34173cb38f07f89ddbebc2ac9128303f-Paper-round2.pdf.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

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 *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9338–9346. IEEE, 2019.

Gershom Seneviratne, Yohan Abeysinghe, Jianyu An, Vaibhav Shende, Rahul Kumar, and Dinesh Manocha. VEGA: Learning navigation VLAs from in-the-wild egocentric video with geometric trajectory supervision. *arXiv preprint arXiv:2606.18426*, 2026.

Dhruv Shah, Błażej Osiński, brian ichter, and Sergey Levine. LM-Nav: Robotic navigation with large pre-trained models of language, vision, and action. In Karen Liu, Dana Kulic, and Jeff Ichnowski (eds.), *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 492–504. PMLR, 14–18 Dec 2023a. URL https://proceedings.mlr.press/v205/shah23b.html.

Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. Gnm: A general navigation model to drive any robot. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7226–7233, 2023b. doi: 10.1109/ICRA48891.2023.10161227.

Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. Vint: A foundation model for visual navigation, 2023c. URL https://arxiv.org/abs/2306.14846.

Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masked diffusion policies for navigation and exploration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 63–70, 2024. doi: 10.1109/ICRA57147.2024.10610665.

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.

Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7464–7475. IEEE, 2023.

Jiahao Wang, Yufeng Yuan, Rujie Zheng, Youtian Lin, Jian Gao, Lin-Zhuo Chen, Yajie Bao, Chang Zeng, Yanxi Zhou, Xiao-Xiao Long, et al. SpatialVID: A large-scale video dataset with spatial annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 42592–42603, 2026a.

Shaoan Wang, Aocheng Luo, Fei Huang, Jingyi Xu, Xiaoyang Wang, Yueyu Wang, Qianli Ma, Fan Yang, Ran Mei, Jia Wei, et al. LightNav-0: Eliciting VLM spatial intelligence for generalist embodied navigation. *arXiv preprint arXiv:2608.30935*, 2026b.

Shihao Wang, Shilong Liu, Yuanguo Kuang, Xinyu Wei, Yangzhou Liu, Zhiqi Li, Yunze Man, Guo Chen, Andrew Tao, Guilin Liu, et al. LocateAnything: Fast and high-quality vision-language grounding with parallel box decoding. In *European Conference on Computer Vision*, pp. 336–357. Springer, 2026c.

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. *arXiv preprint arXiv:2507.05240*, 2025.

Meng Wei, Chenyang Wan, Jiaqi Peng, Xiqian Yu, Yuqiang Yang, Delin Feng, Wenzhe Cai, Chenming Zhu, Tai Wang, Jiangmiao Pang, and Xihui Liu. Ground slow, move fast: A dual-system foundation model for generalizable vision-language navigation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=GK4rznYwhn.

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 *International Conference on Learning Representations*, volume 2024, pp. 10641–10662, 2024.

Ziyuan Xia, Jingyi Xu, Chong Cui, Yuanhong Yu, Jiazhao Zhang, Qingsong Yan, Tao Ni, Junbo Chen, Xiaowei Zhou, Hujun Bao, et al. Habitat-GS: A high-fidelity navigation simulator with dynamic Gaussian splatting. In *European Conference on Computer Vision*, pp. 306–323. Springer, 2026.

Zhuo Xu, Hao-Tien Lewis Chiang, Zipeng Fu, Mithun George Jacob, Tingnan Zhang, Tsang-Wei Edward Lee, Wenhao Yu, Connor Schenck, David Rendleman, Dhruv Shah, Fei Xia, Jasmine Hsu, Jonathan Hoech, Pete Florence, Sean Kirmani, Sumeet Singh, Vikas Sindhwani, Carolina Parada, Chelsea Finn, Peng Xu, Sergey Levine, and Jie Tan. Mobility VLA: Multimodal instruction navigation with long-context VLMs and topological graphs. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 3866–3887. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/xu25b.html.

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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4927–4936. IEEE, 2023.

Ning Yang, Yan Huang, Kaiwen Peng, Ziheng He, Kai Wang, Cui Miao, Kailin Lyu, Guo Li, Xiaofeng Wang, Zheng Zhu, et al. WAM-Nav: Asymmetric latent world-action modeling for unified visual navigation. *arXiv preprint arXiv:2606.04907*, 2026.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. CogVideoX: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, volume 2025, pp. 83048–83077, 2025.

Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 28213–28239, 2025. URL https://proceedings.iclr.cc/pa

per_files/paper/2025/file/45d74e190008c7bff2845ffc8e3facd3-Paper-Conference.pdf.

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.

Tianwei Yin, Qiang Zhang, Richard Zhang, William T Freeman, Fredo Durand, Eli Shechtman, and Xun Huang. From slow bidirectional to fast autoregressive video diffusion models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22963–22974. IEEE, 2025.

Naoki Yokoyama, Ram Ramrakhya, Abhishek Das, Dhruv Batra, and Sehoon Ha. HM3D-OVON: A dataset and benchmark for open-vocabulary object goal navigation. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5543–5550. IEEE, 2024.

Hai Zhang, Siqi Liang, Li Chen, Yuxian Li, Yukuan Xu, Yichao Zhong, Fu Zhang, and Hongyang Li. Sparse video generation propels real-world beyond-the-view vision-language navigation. *arXiv preprint arXiv:2602.05827*, 2026a.

Jiazhao Zhang, Kunyu Wang, Rongtao Xu, Gengze Zhou, Yicong Hong, Xiaomeng Fang, Qi Wu, Zhizheng Zhang, and He Wang. NaVid: Video-based VLM Plans the Next Step for Vision-and-Language Navigation. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, July 2024. doi: 10.15607/RSS.2024.XX.079.

Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-NaVid: A Video-based Vision-Language-Action Model for Unifying Embodied Navigation Tasks. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025a. doi: 10.15607/RSS.2025.XXI.013.

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 *International Conference on Learning Representations*, volume 2026, pp. 127293–127322, 2026b.

Mingjie Zhang, Yuheng Du, Chengkai Wu, Jinni Zhou, Zhenchao Qi, Jun Ma, and Boyu Zhou. Apexnav: An adaptive exploration strategy for zero-shot object navigation with target-centric semantic fusion. *IEEE Robotics and Automation Letters*, 10(11):11530–11537, 2025b. doi: 10.1109/LRA.2025.3606388.

Baining Zhao, Jiacheng Xu, Weicheng Feng, Xin Zhang, Zhaolu Wang, Haoyang Wang, Shilong Ji, Ziyou Wang, Jianjie Fang, Zhiheng Zheng, et al. WorldVLN: Autoregressive world action model for aerial vision-language navigation. *arXiv preprint arXiv:2605.15964*, 2026.

Han Zheng, Zhe Chen, Yiwen Fu, Ming Yang, and Tong Qin. SCAN-Planner: Spatial collision-aware local planning for route-guided long-range quadruped navigation. *arXiv preprint arXiv:2606.19555*, 2026.

Changqing Zhou, Yueru Luo, Zeyu Jiang, and Changhao Chen. UniNav: A unified world-action diffusion model for visual navigation. *arXiv preprint arXiv:2608.03244*, 2026.

Chuning Zhu, Raymond Yu, Siyuan Feng, Benjamin Burchfiel, Paarth Shah, and Abhishek Gupta. Unified World Models: Coupling Video and Action Diffusion for Pretraining on Large Robotic Datasets. In *Proceedings of Robotics: Science and Systems*, LosAngeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.015.

Weiye Zhu, Zekai Zhang, Xiangchen Wang, Hewei Pan, Teng Wang, Tiantian Geng, Rongtao Xu, and Feng Zheng. NaVIDA: Vision-language navigation with inverse dynamics augmentation. *arXiv preprint arXiv:2601.18188*, 2026.

## A Qualitative Analysis of Video Pretraining

Figure 4 compares the original Wan2.2-TI2V-5B model with our model after Stage 1, given the same initial frame and navigation instruction. Wan 2.2 generates entire clips bidirectionally, while our model generates chunks autoregressively. In both examples, Wan 2.2 animates a human subject unrelated to navigation and keeps the camera largely stationary, whereas our model predicts camera motion toward the target.

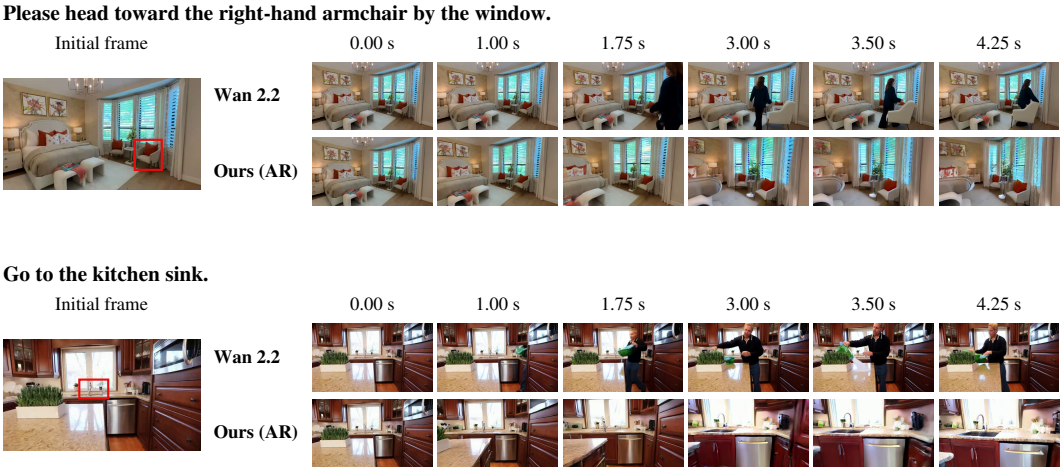



Figure 4: Qualitative comparison of Wan 2.2 and our Stage 1 model with autoregressive (AR) video generation. Red boxes mark the navigation targets.

## B Implementation Details

**Model and input configuration.** We use RGB frames at a resolution of $512 \times 288$ pixels and a sampling rate of 4 Hz. Each video chunk contains nine observations, $o_i, \ldots, o_{i+8}$, with $o_i$ serving as the current-observation condition. Its $T = 9$ actions correspond to the transitions $o_i \to o_{i+1}, \ldots, o_{i+8} \to o_{i+9}$, where $o_{i+9}$ is the first observation of the next chunk. The VAE encodes the nine frames into three temporal latent slices. The history window retains the two most recent chunks in addition to the initial-frame anchor. Each action component $u$ is normalized using a scaled symlog transformation (Hafner et al., 2025), $f(u) = \mathrm{sign}(u)\log(1 + |u|/c)$, with $c = 0.01\,\mathrm{m}$ for translation and $c = 0.01\,\mathrm{rad}$ for yaw. We recover action components using $f^{-1}(z) = c\,\mathrm{sign}(z)(\exp(|z|) - 1)$.

**Training configuration.** We use AdamW (Loshchilov & Hutter, 2019) with BF16 mixed precision throughout training. Table 4 summarizes the stage-specific settings. Video pretraining consists of 2,000 updates under the original video-training formulation, followed by 8,000 updates with chunk-wise causal prediction. For joint fine-tuning, we warm up the learning rate over 500 updates, hold it constant until update 4,000, and apply cosine decay to $5 \times 10^{-7}$ by update 5,500. It remains constant thereafter, and we use the checkpoint at update 6,500 for evaluation.

**Inference and execution.** We jointly generate video latents and action chunks using four Euler denoising steps. The classifier-free guidance (Ho & Salimans, 2022) scales are 5.0 for video and 1.0 for actions, with noise-schedule shift factors of 5.0 and 1.0, respectively. At each denoising step, the model receives the video and action noise levels from their respective schedules, and each modality is updated using its own Euler step size. The unconditional branch removes the text condition while retaining the current observation and history. During simulation evaluation, we execute the nine actions in each chunk sequentially at 4 Hz and predict the next chunk from the latest observation, unless the episode terminates. The visual history is constructed from observations collected during

Table 4: Training configuration. Update counts refer to the checkpoints used in the final model's training sequence. Video replay probabilities specify dataset sampling, not loss weights. The dash indicates that Stage 1 uses only Internet egocentric videos, rather than a replay mixture.

| Configuration | Video pretraining | Action alignment | Joint fine-tuning |
|---|---|---|---|
| Updates | 10,000 | 2,500 | 6,500 |
| Global batch size | 64 | 64 | 32 |
| Peak learning rate | $1 \times 10^{-5}$ | $1 \times 10^{-5}$ | $5 \times 10^{-6}$ |
| Video replay probability | – | 10% | 20% |

execution rather than generated frames. During simulation evaluation, action history stores the model's previous action predictions rather than measured robot displacements.

**Inference optimizations.** Historical tokens use fixed zero-noise timesteps and cannot attend to the current chunk. We therefore compute their keys and values once for each classifier-free guidance branch and reuse them across denoising steps (Ye et al., 2026). We batch the conditional and unconditional forward passes for classifier-free guidance and use CUDA graphs[1] to reduce kernel-launch overhead. We also omit RGB decoding of the predicted video latents because action execution does not require the generated frames. At four denoising steps, these optimizations reduce per-chunk policy-call latency from 1,791 to 684 ms, a $2.6\times$ speedup (Table 5). The step-count sweep measures latency only, not navigation success.

We profile ScaleNav on recorded observations in BF16 using CUDA synchronization and the guidance settings above. Table 5 reports mean per-chunk policy-call latency, including current-observation encoding, completed-chunk history encoding, history KV prefill, denoising, and buffered-action retrieval. Video decoding is included only in the ablation configurations that retain it. Measurements exclude simulator rendering, network communication, and motion execution.

The four-step configurations use 23 measured chunks after five warm-up chunks per configuration. Configurations with more denoising steps use 10 measured chunks after four warm-up chunks. Over the 23 measured chunks, the fully optimized four-step implementation has a sample standard deviation of 0.5 ms and an empirical 95th percentile of 685.1 ms for per-chunk policy-call latency.

The baseline profiling scripts time policy calls, including preprocessing and model inference, as well as object grounding, segmentation, semantic fusion, map updates, and planning where applicable. Simulator rendering and motion execution are excluded.

Table 5: Mean per-chunk policy-call latency on an RTX 4090 (ms), including history updates. **Left:** unoptimized and fully optimized implementations at different denoising step counts. **Right:** cumulative optimizations at four steps. Speedups are relative to the unoptimized implementation at the same step count.

| Steps | Unoptimized | Optimized | Speedup |
|---|---|---|---|
| 4 | 1791 | **684** | 2.6× |
| 8 | 3272 | 1034 | 3.2× |
| 20 | 7731 | 2080 | 3.7× |
| 50 | 18846 | 4693 | 4.0× |

| Optimization (cumulative) | Latency | Speedup |
|---|---|---|
| Unoptimized | 1791 | 1.0× |
| + History KV caching | 967 | 1.9× |
| + Batched guidance | 948 | 1.9× |
| + CUDA graphs | 920 | 1.9× |
| + Omit video decoding | **684** | 2.6× |

[1] https://docs.nvidia.com/cuda/cuda-programming-guide/04-special-topics/cuda-graphs.html

# C Data Construction Details

We use Gemini 3.5 Flash[2] for VLM-based annotation and instruction paraphrasing. Figure 5 shows examples from the two training datasets.

**Internet egocentric video**



"Go to the wooden kitchen cabinets."

**Action-labeled simulated data**



"Navigate to the blue chair next to the refrigerator visible in your initial view."

Figure 5: Examples from the two training datasets, paired with their navigation instructions. Frames are shown in temporal order. Red boxes mark the targets for illustration only.

## C.1 Internet Egocentric Videos

**Segment extraction.** For each source clip, we sample keyframes at uniform intervals and provide them in temporal order to the VLM. The extraction prompt takes keyframes labeled with source-frame indices and returns segment boundaries and instructions. We map the returned indices to timestamps in the source clip. Initial instructions follow the form "go to ..." and describe the selected target by its category, attributes, or spatial relations.

```
Identify navigation segments from the supplied egocentric video
keyframes. The keyframes are sampled at uniform intervals,
presented in temporal order, and labeled with source-frame
indices.

For each segment:
1. Identify a frame in which the camera has arrived near a
clearly describable object or region. Treat this frame as a
candidate endpoint and the corresponding object or region as the
navigation target.
2. Search backward through the supplied keyframes and identify
the earliest keyframe in which the same target is visible. Use
this as the starting frame. The start must precede the endpoint.
3. Generate a concise navigation instruction in the form "Go to
...", describing the target by its category, attributes, or
spatial relations. Ground the description in the initial
observation.
```

[2] https://ai.google.dev/gemini-api/docs/models/gemini-3.5-flash

```
Return JSON only, with a "candidates" list. Each entry must
contain:
"target_description", "start_frame", "end_frame", and
"instruction". Use only the supplied frame indices. Return
{"candidates": []} if no valid segment is identified.
```

**Segment review.** The segment-review prompt below assesses target-instruction consistency, motion plausibility, arrival accuracy, and temporal continuity. We separately label the candidate segment and the surrounding endpoint context to support arrival assessment.

```
Review the candidate navigation segment and its instruction.

Inputs:
- Instruction: {instruction}
- Target description: {target_description}
- Segment boundaries: {start_frame}, {end_frame}
- Chronologically ordered frames covering the segment.
- Separately labeled source frames around the candidate
endpoint.

Assess the following four criteria:
1. Target-instruction consistency: Verify that the initial and
final frames depict the same target and that its category and
visible attributes match the instruction. Check that any stated
spatial relations are supported by the initial observation,
using that view as the reference for left/right and ordinal
descriptions. When multiple plausible targets are visible,
verify that the instruction uniquely identifies the intended
target. Mark this criterion as "fail" if the wording is
ambiguous, or "uncertain" if the visual evidence is insufficient
to decide.
2. Motion plausibility: Check that the segment primarily shows
camera movement toward the target, rather than prolonged
in-place rotation, stationary observation, or back-and-forth
motion that does not advance toward it.
3. Arrival accuracy: Examine the frames around the candidate
endpoint to determine whether the camera has reached the
target's vicinity.
4. Temporal continuity: Check for scene cuts or abrupt temporal
jumps within the selected segment.

Return JSON with the keys "target_instruction_consistency",
"motion_plausibility", "arrival_accuracy", and
"temporal_continuity". For each criterion, provide a "decision"
("pass", "fail", or "uncertain") and a brief "reason" grounded
in the supplied frames. Include "keep", set to true only when
all four criteria pass.
```

### C.2 Action-Labeled Simulated Data

**Scene and target selection.** Our candidate pool contains approximately 1K InteriorGS scenes (Miao et al., 2026), each with instance-level semantic 3D bounding boxes and room layout annotations. We exclude scenes with poor rendering quality or inadequate semantic annotations. We randomly sample starting poses from navmesh-defined traversable areas. Given the initial observation and scene annotations, we select a target and its corresponding goal position. We then form an initial instruction from the target's category or spatial relation and refine ambiguous references with the prompt below.

**Trajectory generation and action labels.** Our offline 2D adaptation of SCAN-Planner (Zheng et al., 2026) uses A* (Hart et al., 1968) to guide local obstacle avoidance and optimizes the path while accounting for the robot's footprint. The planner then performs time reparameterization and collision checking. We sample egocentric observations and robot poses at 4 Hz while executing the time-parameterized trajectory. We convert consecutive global poses into relative poses in the preceding robot body frame to obtain action labels.

**Instruction refinement.** The following prompt refines ambiguous target references without changing the selected target or destination.

```
Refine the navigation instruction to distinguish the selected
target from other objects of the same category visible in the
initial observation.

Inputs:
- Initial egocentric observation.
- Initial instruction: {instruction}
- Selected target: {selected_target}
- Spatial and semantic information for the relevant objects:
  {object_annotations}

Preserve the selected target and intended destination. When the
category name alone is ambiguous, add a concise description of
the target's visible attributes or relative position, such as
"the red chair on the right". Interpret directional descriptions
from the initial observation.

Use the scene annotations to associate the selected target with
the image. Include only attributes and spatial relations
supported by the initial observation. Do not change the target,
invent distinguishing features, or introduce additional
navigation requirements.

If the original instruction already identifies the intended
target unambiguously, retain it.

Return JSON with:
- "instruction": the refined instruction, or null if the target
cannot be distinguished using the supplied information.
- "reason": a brief explanation of the distinguishing visual
evidence.
```

**Review with scene annotations.** The following addendum supplements the segment-review prompt with scene annotations and a target-marked reference image.

```
Apply the segment-review criteria to the supplied simulated
clip. For arrival accuracy, use the final observation and nearby
frames from the rendered sequence.

Additional inputs:
- Scene semantic annotations: {scene_annotations}
- A target-marked reference image identifying the intended
target.
```

```
When assessing target-instruction consistency, combine these
inputs with the unmarked observations to verify that the
instruction refers to the selected target. Check that its
category, distinguishing attributes, and stated spatial
relations are supported by the initial observation. The
reference marker identifies the intended target and is not a
property of the object or part of the navigation instruction.

Use the same four assessment keys and output format as the
segment-review template. Include the additional evidence in the
reason for "target_instruction_consistency".
```

### C.3 Instruction Paraphrasing

We use the following prompt to paraphrase accepted instructions from both datasets.

```
Paraphrase the accepted navigation instruction to increase
language diversity while preserving its meaning.

Original instruction: {instruction}

Preserve the intended target, destination, distinguishing
attributes, and spatial relations. Preserve the directional
references in the original instruction. Do not introduce new
navigation requirements or remove information needed to identify
the intended target.

Return JSON only, with the paraphrased instruction under
"instruction".
```

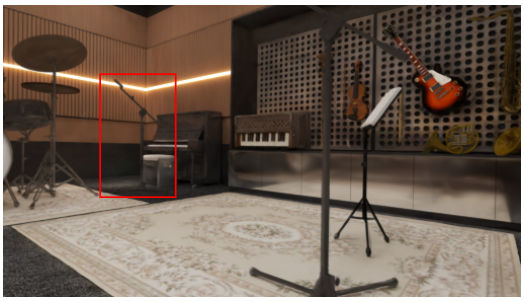
## D Benchmark and Evaluation Details

**Single-view adaptation of Short-Horizon OVON.** To adapt Short-Horizon OVON (Gong et al., 2026) to our single-view setting, we retain episodes with an estimated target visibility of at least 50% in the initial front view. We estimate visibility as the fraction of sampled vertices from the target's semantic mesh (Yadav et al., 2023) whose projections fall within the same instance's region in the initial front-view semantic image. This is a target-visibility fraction, not the fraction of image pixels occupied by the target. The resulting evaluation set contains 524 episodes across 36 HM3D scenes (Ramakrishnan et al., 2021), each paired with a navigation instruction referring to the target object. Because this evaluation uses our single-view selection and the shared protocol in Section 4.1, its results are not directly comparable to results under the original three-view protocol.

**HabitatGS-LN construction.** We use indoor scenes from InteriorGS (Miao et al., 2026) that are held out from training. Their scene IDs are disjoint from those of the 650 scenes used for our 2,600 action-labeled simulated training episodes. Using the simulation pipeline described in Section 3.3, we generate candidate episodes covering diverse object categories and spatial descriptions. Each episode specifies an initial robot pose and a navigation instruction that unambiguously refers to a specific object visible in the initial observation, rather than an object category. Although our task formulation allows both object and region targets, we restrict this benchmark to object targets for comparison with baselines designed for object-goal navigation.

**Human review.** To ensure the quality and validity of the evaluation tasks, two human reviewers independently inspect each candidate episode. They assess instruction accuracy and clarity, target visibility in the initial observation, destination reachability, and scene rendering quality. We retain only episodes approved by both reviewers, yielding 157 episodes across 45 scenes.

Figure 6 shows example tasks from HabitatGS-LN.



"please move to the microphone on your front-left visible in your initial view"



"Head to the massage chair ahead visible in your initial view."

Figure 6: Example tasks in HabitatGS-LN with their final navigation instructions. Red boxes mark the specified targets.

**Simulation camera configuration.** For all methods on both simulation benchmarks, we render RGB observations at a resolution of $512 \times 288$ pixels with a horizontal field of view of approximately $70°$.

**Simulation action execution.** In both environments, we conduct closed-loop evaluation by executing the predicted actions and updating observations as the robot moves. We convert each method's discrete or continuous action outputs into kinematic pose updates. Within each scene, all methods use the same navmesh-based motion constraints.

**NavDP goal construction.** We compute the median of the valid depth values within the detected bounding box and select the pixel whose depth is closest to this median. Using its depth, the camera intrinsics, and the simulator-provided camera pose, we back-project this pixel into world coordinates without snapping the resulting point to the navmesh. We keep this world-frame goal fixed and express it as a 2D vector in the robot's current local frame at each step. NavDP predicts motion trajectories conditioned on this local goal and RGB-D observations.

**ApexNav adaptation.** ApexNav's original detection pipeline uses category-level queries. We replace YOLOv7 (Wang et al., 2023) and Grounding DINO (Liu et al., 2024) with LocateAnything to process the referring expressions in our instructions. The resulting detections feed into its segmentation, semantic fusion, and navigation pipeline.

**Baseline fine-tuning.** We prepare fine-tuning data for StreamVLN and Uni-NaVid from our action-labeled simulated training episodes. A path follower converts the reference trajectories into each model's native discrete actions. We execute these actions in Habitat-GS and re-render observations at the corresponding poses to ensure alignment between the visual observations and discrete action labels. We confirmed that all trajectories used for fine-tuning terminated near their intended goals. The resulting observation-action sequences are then paired with the navigation instructions and formatted for each model.

We adopt the optimization hyperparameters from StreamVLN's Stage 1 training script and Uni-NaVid's Stage 2 fine-tuning script. We initialize both models from their released checkpoints and fine-tune each model for five epochs using AdamW with a global batch size of 128. StreamVLN uses a base learning rate of $2 \times 10^{-5}$ and a learning rate of $5 \times 10^{-6}$ for its vision encoder. Uni-NaVid uses a learning rate of $10^{-5}$ and keeps its vision encoder frozen.

**Footprint distance.** The robot and target footprints are the regions they occupy when projected onto the ground plane. We model the robot footprint as an oriented capsule with a radius of 0.25 m and a centerline length of 0.36 m, using SCAN-Planner's Go2 parameters (Zheng et al., 2026). We measure the minimum Euclidean distance between this capsule and the target's annotated ground-plane polygon (Ericson, 2004).

**SPL reference lengths.** We compute reference lengths as the shortest navmesh distances to annotated goal positions on HabitatGS-LN and to sampled navigable points near valid target footprints on Short-Horizon OVON.

**Stopping rules.** Our model stops when five consecutive actions have translation magnitudes below 0.015 m and absolute yaw changes below 0.015 rad. NavDP stops when the robot is within 1 m of its estimated goal point on the ground plane. For the remaining baselines, we retain their default stopping mechanisms: explicit STOP actions for StreamVLN and Uni-NaVid, and rule-based stopping for ApexNav.

**Pseudo-action labeling ablation.** We use Depth Anything 3 (DA3-Nested) (Lin et al., 2026) to estimate metric camera poses from Internet egocentric videos. We invert the predicted world-to-camera transforms to obtain camera-to-world poses. We then express consecutive relative motions in a camera-aligned planar frame to obtain pseudo action labels. We retain clips with available pose estimates and do not apply confidence-based loss weighting.

## E REAL-ROBOT EVALUATION

**System.** We deploy the policy on a Unitree Go2 equipped with a front-mounted Intel RealSense D435i camera and an onboard Jetson computer. The client sends rectified $512 \times 288$ RGB observations to a remote RTX 4090 GPU for inference. For both ScaleNav and the variant without Internet egocentric videos, we convert each predicted motion increment $a_t = (\Delta x_t, \Delta y_t, \Delta \psi_t)$ into body-frame linear and angular velocity references:

$$v_t^x = \frac{\Delta x_t}{\Delta t}, \qquad v_t^y = \frac{\Delta y_t}{\Delta t}, \qquad \omega_t = \frac{\Delta \psi_t}{\Delta t},$$

where $\Delta t = 0.25$ s is the time interval represented by each action. These velocity references are passed to the robot's low-level motion interface for execution. After completing each action chunk, the robot holds still until the next prediction is available.

StreamVLN and Uni-NaVid predict discrete forward, left-turn, right-turn, and STOP actions. Forward actions specify a 0.25 m translation, while left/right turns specify $15^\circ$ rotations for StreamVLN and $30^\circ$ rotations for Uni-NaVid. All reported baseline trials use step-wise execution. The client uses robot odometry to measure progress and adjusts linear or angular velocity according to the remaining distance or yaw error. After completing an action, it acquires a new RGB observation before proceeding to the next action. A predicted STOP action terminates navigation.

**Tasks and results.** Figure 7 presents the initial views and verbatim task instructions, and Table 6 reports per-task success counts. All methods use an execution budget of 200 actions per trial.

Table 6: Per-task real-robot results, reported as successes / trials. Best results in each row are in **bold**.

| Task | StreamVLN (fine-tuned) | Uni-NaVid (fine-tuned) | Ours (w/o Internet videos) | Ours |
|---|---|---|---|---|
| *Hall* | | | | |
| Display counter | 2/5 | 4/5 | 4/5 | **5/5** |
| Rocket model | 1/5 | 1/5 | 2/5 | **3/5** |
| Screen on the rack | 0/5 | 4/5 | **5/5** | **5/5** |
| Second door | 2/5 | 3/5 | **4/5** | **4/5** |
| *Lounge* | | | | |
| Wooden cabinet | 0/5 | 0/5 | 1/5 | **3/5** |
| Leftmost white cylinder | 3/5 | **5/5** | 4/5 | 4/5 |
| Second table | 4/5 | **5/5** | 3/5 | **5/5** |
| Vending machine | **5/5** | **5/5** | 4/5 | **5/5** |
| *Office* | | | | |
| Cabinet under the printer | 0/5 | 0/5 | 1/5 | **3/5** |
| Left trash can | 0/5 | 4/5 | **5/5** | **5/5** |
| Glass door | 4/5 | 0/5 | 4/5 | **5/5** |
| Tripod phone stand | **4/5** | 0/5 | 2/5 | 3/5 |

**Hall**

**(a) Display counter**



"Head to the large white display counter on your left visible in your initial view."

**(b) Rocket model**



"Make your way to the rocket model visible in your initial view."

**(c) Screen on the rack**



"Walk over to the freestanding screen on the equipment rack visible in your initial view."

**(d) Second door**



"Walk over to the second door on your left visible in your initial view."

**Lounge**

**(e) Wooden cabinet**



"Walk over to the wooden cabinet on your left visible in your initial view."

**(f) Leftmost white cylinder**



"Go over to the leftmost short white cylinder visible in your initial view."

**(g) Second table**



"Move toward the second table on your left visible in your initial view."

**(h) Vending machine**



"Proceed to the vending machine visible in your initial view."

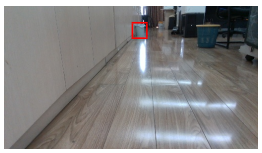
**Office**

**(i) Cabinet under the printer**



"Go to the wooden cabinet under the printer visible in your initial view."

**(j) Left trash can**



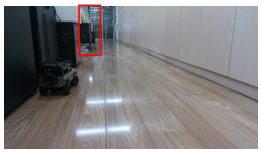
"Reach the trash can on the left visible in your initial view."

**(k) Glass door**



"Walk up to the glass door on your left visible in your initial view."

**(l) Tripod phone stand**



"Approach the tripod phone stand visible in your initial view."

Figure 7: Real-robot task views and instructions, grouped by environment. Red boxes mark the targets. Target boxes and privacy masks are for illustration only and are not policy inputs.