MM-Nav: Multi-View VLA Model for Robust Visual Navigation via Multi-Expert Learning

Tianyu Xu[1,2*], Jiawei Chen[1*], Jiazhao Zhang[1,2*], Wenyao Zhang[2,3], Zekun Qi[2,4], Minghan Li[2], Jiahang Liu[1,2], Lu Yue[1,2], Zhizheng Zhang[2†], and He Wang[1,2†]

[1] Peking University, Beijing, China
[2] Galbot, Beijing, China
[3] Shanghai Jiao Tong University, Shanghai, China
[4] Tsinghua University, Beijing, China

Abstract. Visual navigation policies are widely regarded as a critical research direction, as they emulate human navigation behavior by leveraging egocentric visual observations. However, unlike LiDAR point clouds or depth maps, visual observations do not provide explicit geometric information for navigation, especially in cluttered or dynamic environments, which motivates the need for learning-based models and large-scale data. To this end, we propose to leverage Vision-Language-Action (VLA) models to learn diverse navigation capabilities from synthetic expert data and to alleviate the sim-to-real gap by co-training on large-scale real-world Visual Question Answering (VQA) data. Specifically, we develop MM-Nav, a 7B multi-view VLA model featuring custom-designed architectures and enabling a 7 Hz inference speed with 360° observation. For large-scale navigation data, we collect a total of 1.5 million expert demonstrations from three reinforcement learning (RL) experts, each trained with privileged information in a challenging, tailor-made environment and specialized in one of three navigation capabilities: *reaching*, *squeezing*, and *avoiding*. We then iteratively train MM-Nav on these data, dynamically balancing the training data ratio across the three capabilities based on their respective performance. Through extensive experiments in synthetic and real-world environments, we demonstrate that our model achieves strong performance and generalization on different benchmarks. MM-Nav obtains a success rate of 88.1% on the InternVLA-N1 System-1 point-goal navigation benchmark. Moreover, we find that our student VLA model outperforms the RL teachers, demonstrating the synergistic effect of integrating multiple capabilities. Extensive real-world experiments further confirm the effectiveness of our method.

Keywords: Visual navigation policy · Vision-Language-Action model

*: Equal contributions, †: Corresponding authors.
Project page: `https://pku-epic.github.io/MM-Nav-Web/`

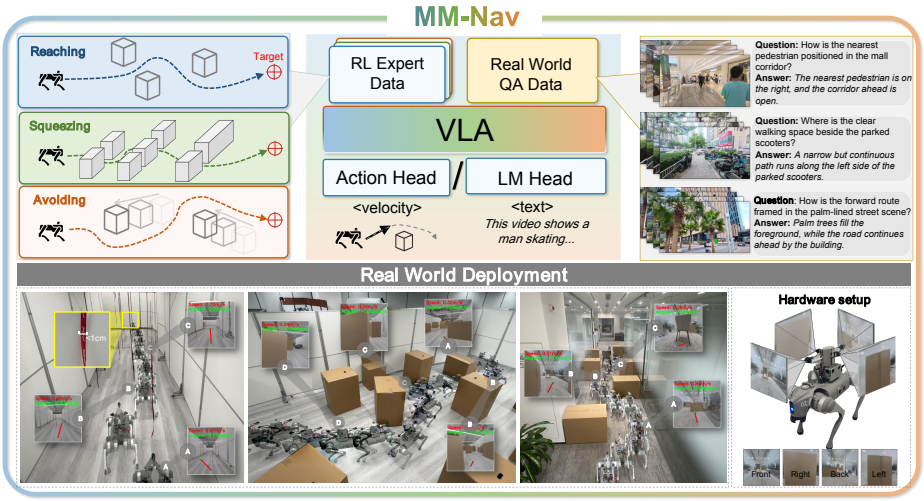


Fig. 1: MM-Nav is a multi-view VLA policy trained from capability-specific RL expert trajectories together with large-scale QA data, and outputs continuous velocity commands via an action head. **Bottom-left**: real-world rollouts illustrating robust navigation in challenging scenes, including thin-obstacle (wire) avoidance and squeezing through cluttered/transparent structures. **Bottom-right**: robot platform and the 360° surround-view camera configuration used for deployment.

1 Introduction

Visual navigation has garnered considerable attention in the robotics field [19, 23,34,49,50,54,58,59,64,78,79,84], as it requires robots to reach target locations based on visual inputs. Visual observations provide detailed and rich environmental information for navigation while remaining cost-effective. However, in complex and cluttered navigation environments, interpreting such informative visual data and planning appropriate navigation actions remains challenging [50,54,58], demanding highly intelligent models and large-scale navigation datasets.

To this end, existing methods [49,50,54,59,78] learn visual navigation policies from synthetic data or teleoperated demonstrations, with recent works further leveraging real-world passive navigation videos to emulate general navigation behaviors [4, 16, 17, 30].However, real-world data collection is expensive and often restricted to limited camera and scene configurations [5,37] (*e.g.*, front-view only). While synthetic data can be generated more flexible and efficiently, it typically suffers from significant sim-to-real gaps due to non-photorealistic rendering [2, 35, 57]. Moreover, both paradigms are largely confined to relatively spacious environments [2, 9] and rarely include highly challenging or hazardous scenarios. This is because collecting real-world data for collision avoidance is costly and risky [26, 43], while generating such data in simulation through rule-based approaches remains nontrivial [41, 46, 88].

To address these limitations, we propose MM-Nav, a multi-view Vision-Language-Action (VLA) model that captures 360° observations around the robot, learns navigation capabilities from multiple synthetic RL experts, and mitigates

the sim-to-real gap by co-training on large-scale real-world Visual Question Answering (VQA) data, as shown in Figure 1. To overcome the lack of highly challenging scenarios in existing navigation datasets, we construct synthetic environments and train separate RL experts for three distinct skills: reaching, squeezing, and avoiding, which are difficult and risky to obtain in the real world. We collect 500k successful expert demonstrations to form a capability-diverse training dataset that is challenging to obtain in real-world settings. Specifically, we first initialize MM-Nav on the collected expert demonstrations and then iteratively refine it with an online teacher-student strategy in a DAgger manner [42]. Beyond standard DAgger, we dynamically balance the training data ratio across the three capabilities based on their respective performance, which facilitates stable improvement and accelerates convergence. Moreover, building upon a VLM framework [77], we co-train MM-Nav on 300k real-world VQA data. This enables the model to leverage the real-world visual understanding from VQA, thereby strengthening its generalization ability and mitigating the sim-to-real gap.

We conduct extensive experiments in both synthetic and real-world environments. The results demonstrate that MM-Nav achieves strong navigation performance across diverse capability settings, even outperforming specifically trained RL experts. Notably, on the InternVLA-N1 System-1 point-goal navigation benchmark [22], MM-Nav achieves an average SR of 88.1%, outperforming the best previous method (72.2%) by +15.9 points. Extensive real-world evaluations further confirm its robust zero-shot sim-to-real transfer ability in challenging environments. In addition, our efficient tokenization design enables inference at approximately 7 Hz, comparable to existing visual navigation methods [50,78].

The primary contributions can be summarized as follows: (1) *Navigation data*: we build tailored multi-capability simulation environments and train privileged RL experts to provide diverse supervision for reaching, squeezing, and dynamic obstacle avoidance. (2) *Training strategy*: we introduce a capability-balanced iterative refinement procedure that mitigates policy-distribution shift and alleviates capability imbalance across different navigation skills. (3) *Model design*: we develop an efficient 7B multi-view VLA policy that uses 360° RGB observations and directly predicts continuous omnidirectional velocity at 7 Hz.

2 Related Works

Learning-based small navigation model. A considerable amount of learning-based navigation relies on models with a smaller parameter count. These models are often computationally efficient for specific tasks but face limitations in generalization and adaptability to complex scenarios. They can be broadly categorized into imitation learning and reinforcement learning. Imitation Learning (IL) acquires navigation capabilities by learning from expert demonstrations. This approach is relatively data-efficient as the expert data inherently contains effective strategies [1, 2, 8–10, 18, 19, 23, 33, 50, 54, 63, 65]. For instance, NavDP [2] and LoGoPlanner [38] utilizes an A* planner with access to privileged information to generate high-quality expert trajectories. Reinforcement Learning (RL) offers a structured approach to end-to-end navigation by leveraging modern simulation

technologies [12, 14, 41, 61, 66, 72, 75]. These methods are often limited to model size and face strong difficulties in handling the sim-to-real gap. In contrast, our method leverages VLA, utilizing VLA's inherent generalization capabilities to address the sim-to-real transfer challenge.

Learning-based large navigation model. Large model learning [3,6,15,22,24, 28, 55, 60, 67, 71, 77, 80, 82, 87], particularly with Vision-Language-Action models, has significantly advanced navigation by leveraging powerful generalization and semantic understanding capabilities [3, 4, 21, 40, 52, 59, 73, 74, 78, 79]. An approach employs off-the-shelf large models in a zero-shot manner [31,32,47,48], converting visual scenes into text descriptions for a Large Language Model to perform high-level planning [20, 39, 53, 85, 86]. However, this vision-to-text abstraction creates an information bottleneck, often limiting the agent to sparse landmarks and static environments. End-to-end VLA models like NaVid [79] and Uni-NaVid [78] tackle tasks ranging from object-search navigation and human-following to open-ended instruction following, all through a unified language-guided policy. In contrast to these models, which are constrained by a discrete action space (e.g., FORWARD, TURN LEFT) [78,79], our VLA policy directly outputs continuous velocity commands, enabling the agile and instantaneous responses required for robust real-world deployment.

Visual-only navigation. Visual-only navigation relies solely on RGB images to guide movement, which is more economical and lightweight for practical deployment. A primary challenge for such methods is the implicit nature of 3D geometry in 2D images, as pure RGB input lacks direct depth information, making robust obstacle avoidance non-trivial [70]. Foundational models like GNM [49], ViNT [50], Scaling [34], and NoMaD [54] have demonstrated impressive generalization through imitation learning on large-scale datasets, but their reliance on single-camera views limits their spatial awareness. To mitigate safety risks, CARE [23] estimates RGB depth to add a collision avoidance layer on top of a pre-trained policy. Our work addresses these limitations using a four-camera surround-view system for 360° perception [19]. To tackle the challenge of learning from depthless RGB data, we leverage the comprehensive ability of VLA. We adopt the policy distillation paradigm, proven effective in works like X-Nav [56] and COMPASS [29]. But with a key innovation: we train multiple RL experts on specific capabilities using privileged depth information. This expertise is then distilled into a unified VLA policy that robustly navigates by inferring environmental geometry from surround-view RGB images only.

3 Method

3.1 Overview

Task Definition. We formulate the task as learning a velocity control policy π for an omnidirectional robot to navigate safely to a specific point goal in a cluttered and dynamic environment. At each time step t, given the position of a point goal $g_t = [g_t^{\mathrm{x}}, g_t^{\mathrm{y}}]$ and online captured multi-view RGB frames $O_t = \{I_{t-k+1:t}^N\}$ where $I_t^N \in \mathbb{R}^{3\times H\times W}$ (N denotes the number of cameras, including 4 camera views $\{I^{\mathrm{front}}, I^{\mathrm{right}}, I^{\mathrm{back}}, I^{\mathrm{left}}\}$), the policy $\pi(O_t, g_t) \mapsto a_t$ predicts an action

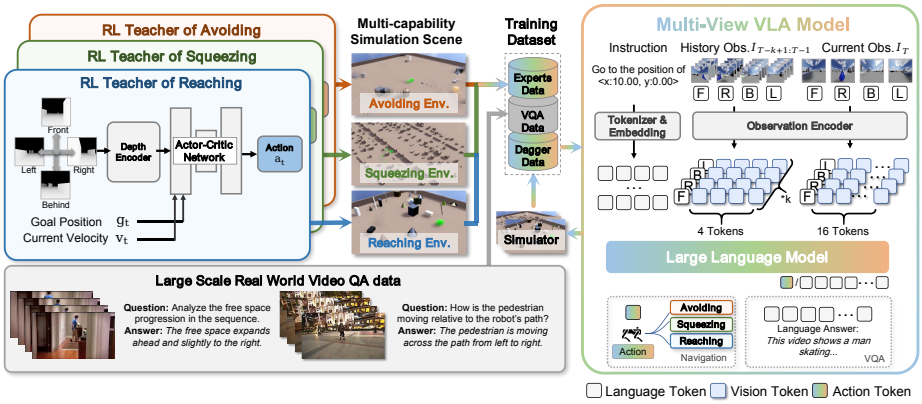


Fig. 2: Pipeline of MM-Nav. Our proposed teachers-student training pipeline. Independent RL teachers are trained in different scenes for multi-capability and distill knowledge to the VLA student. Further, the student is deployed in the capability-specific simulation scene to be iteratively fine-tuned.

$a_t = [v_t^{\mathrm{x}}, v_t^{\mathrm{y}}, v_t^{\mathrm{yaw}}]$, representing omnidirectional velocity. We adopt velocity control because it allows the robot to execute arbitrary omnidirectional motions on a two-dimensional plane, thus enhancing its responsiveness in obstacle avoidance [29,66,72]. The objective is to ensure the velocities generated by the policy π are collision-free and reach the designated goal.

Pipeline Overview. Our pipeline comprises two steps: (1) training of multiple RL experts with different capabilities and initial VLA finetuning; (2) teachers-student online training iteration between RL experts and VLA. As shown in Figure 2, we first use reinforcement learning to train three capability-specific experts in simulation, including reaching, squeezing, and avoiding. These capabilities are obtained specifically by training privileged RL experts in tailored scenarios. Second, different capability navigation trajectories from RL experts are generated and collected, which are then used to directly train a VLA model to initialize a general navigation policy mastering different capabilities. Finally, the basic VLA model (student) is deployed in the simulation environment, and we online collect the expert action of the RL teachers for further finetuning the VLA model. This mechanism is conducted iteratively until performance converges.

3.2 RL Experts for Different Navigation Capabilities

We construct three distinguishing environments for training RL experts to obtain different navigation capabilities: (1) reaching: approach and reach a specific point goal while avoiding static obstacles, (2) squeezing: squeeze through cluttered and narrow gaps between obstacles and walls, and (3) avoiding: actively avoid crowded dynamic obstacles moving at random speed, as illustrated in Figure 3.

Reaching environments. We construct the reaching scene which contains randomized static obstacles, generated in different shapes (cuboids, cones, cylinders, long poles, etc.), textures, and sizes. As in [66], the high diversity of the scene improves the expert's generalization ability, as well as the diversity of the collected

data. Robots and their corresponding goals are initialized at random positions on the terrain, with initial distances of up to 30 m.

Squeezing environments. The static narrow-squeezing scene consists of densely placed pillars randomly distributed across the terrain and walls with narrow, randomized gaps. The expert must navigate safely and smoothly through these passages using visual feedback from four surround-view cameras. For example, observations from the left and right cameras help determine whether the robot can pass between adjacent obstacles.

Avoiding environments. The dynamic avoidance scene contains densely placed dynamic obstacles moving at velocities between 0.5 m/s and 1.5 m/s, requiring the RL expert to actively avoid collisions. These dynamic obstacles exhibit diverse sizes and geometries, such as spheres, cubes, rods, and cones. Similarly to the reaching scene, the distances between the robot and the target are sampled with values up to 10 m, and no collision occurs when the robot is initialized.

Simulation setup. We construct our environments based on IsaacLab [36] for its modular design and strong reinforcement learning support. Following [2], in all three scenes, the robot is abstracted as a cuboid of size $[0.70\text{ m}, 0.35\text{ m}, 0.50\text{ m}]$ in simulation to improve computational and rendering efficiency.

RL experts architecture. Although trained in three different scenes, all RL experts share a similar training methodology. We employ the Proximal Policy Optimization (PPO) [45] algorithm in simulation with 128 parallel robots. The observation at each time step t is defined as:

$$O_t^{\mathrm{RL}} = [d_t^{\mathrm{front}}, d_t^{\mathrm{right}}, d_t^{\mathrm{back}}, d_t^{\mathrm{left}}, a_{t-1}, g_t], \tag{1}$$

where $d_t^{\mathrm{front}}, d_t^{\mathrm{right}}, d_t^{\mathrm{back}}, d_t^{\mathrm{left}}$ are the depth images from four cameras mounted on the respective sides of the robot, a_{t-1} is the last applied action and g_t is the relative position of the goal. As demonstrated in Figure 2. Each depth image from the four views is encoded into a feature vector by ResNet-18 [13], which is concatenated with the last action a_{t-1}, the goal position g_t, and a history token from the last hidden layer of a three-layer MLP. The concatenated features are fed to the MLP to predict the velocity action $a_t = [v_t^{\mathrm{x}}, v_t^{\mathrm{y}}, v_t^{\mathrm{yaw}}]$. The last hidden state of the MLP is stored and passed to the next time step as the history token. The output a_t is then multiplied and clipped within the maximum velocity limits $v_{\max} = [1.5\text{ m/s}, 1.0\text{ m/s}, \pi/4.0\text{ rad/s}]$:

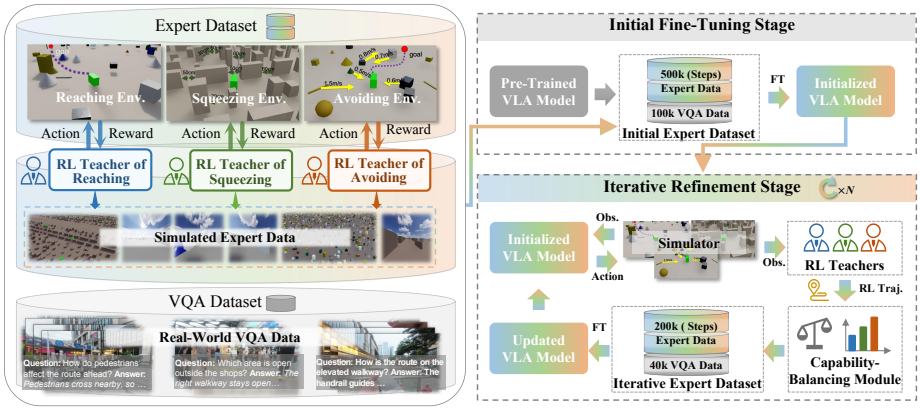
$$v_t = \max\{\min\{a_t, 1.0\}, -1.0\} \times v_{\max}. \tag{2}$$

The resulting velocity v_t is then applied in the simulation. During training, Gaussian noise is added to proprioceptive velocity observations to improve robustness.

RL training rewards and termination conditions. The reward encourages reasonable, goal-directed, and collision-free behavior:

$$r_{\mathrm{Cap.}} = \alpha_{\mathrm{Cap.}} r_{\mathrm{goal}} + \beta_{\mathrm{Cap.}} r_{\mathrm{step}} + \gamma_{\mathrm{Cap.}} r_{\mathrm{reg}} + \delta_{\mathrm{Cap.}} r_{\mathrm{col}}, \tag{3}$$

where Cap. denotes the reaching, squeezing, and avoiding experts; r_{goal} rewards progress toward the goal; r_{step} penalizes each step to encourage efficiency; r_{reg}



(a) Data composition. (b) Different dataset collection stages.

Fig. 3: Training data composition and collection stages of MM-Nav. (a) Dataset composition. MM-Nav is trained on simulated expert trajectories from three RL teachers (reaching, squeezing, avoiding) together with real-world VQA data. (b) Dataset collection stages. The VLA model is initialized with successful teacher demonstrations and real-world VQA data, then iteratively deployed in three environments to collect teacher-supervised trajectories, with a capability-balancing module adjusting data ratios for fine-tuning.

regularizes actions to discourage undesirable behaviors (e.g., retreating, circling); and r_{col} penalizes collisions with obstacles or other robots. To guide and specialize the three RL experts, their reward coefficients $\alpha_{\text{Cap.}}, \beta_{\text{Cap.}}, \gamma_{\text{Cap.}}, \delta_{\text{Cap.}}$ are designed to differ. Specifically, we set $\beta_{\text{Cap.}} = -0.05$ and $\delta_{\text{Cap.}} = -15$ for all experts. For the regularization terms, we assign $\gamma_{\text{reaching}} = 0.05$, $\gamma_{\text{squeezing}} = 0.02$, $\gamma_{\text{avoiding}} = 0$, thereby enforcing stronger regulation in the relatively easier reaching scene while applying weaker or no regularization in the more challenging squeezing and avoiding scenes. Finally, for the goal-reaching reward, we set $\alpha_{\text{reaching}} = \alpha_{\text{avoiding}} = 1.2$ and $\alpha_{\text{squeezing}} = 1.5$, ensuring that the squeezing expert remains decisive when traversing narrow gaps. More details are provided in the supplementary material.

3.3 Student VLA Model

Visual Observation Encoding. To support a 360° observation, we extend a video-based VLA model [79] to support multi-view observations. Given a sliding window (length $k = 8$) of navigation history of four-view RGB images $O_T = I^N_{T-k+1:T}$, which include a front-view, right-view, back-view, and left-view of robots. We then encode all frames into a sequence of visual tokens $E^{\text{visual}} \in \mathbb{R}^{P\times C}$ ($P = 576$ is the patch size and C is the token dimensions) using a visual foundation model (implemented with SigLIP [76]) and a cross-modal projector (a two-layer MLP [28]). Similar to previous methods [78, 79], we conduct grid-based average pooling of the visual tokens, and use fine-grained visual tokens $E^{\text{fine}} \in \mathbb{R}^{16\times C}$ for latest observation and coarse-grained visual tokens $E^{\text{coarse}} \in$

$\mathbb{R}^{4\times C}$ for navigation history. Finally, we can organize the visual token as:

$$E_{\text{visual}} = \{E_{T-k+1}^{\text{F/R/B/L\_coarse}}, ..., E_{T-1}^{\text{F/R/B/L\_coarse}}, E_{T}^{\text{F/R/B/L\_fine}}\}, \tag{4}$$

where all four camera views (Front, Right, Back, Left) are used at each timestep. The use of a sliding window for token selection helps maintain a reasonable visual token sequence length (192 tokens), thereby ensuring consistent inference speed.

Action Forwarding. With organized visual tokens, we format the relative point goal g_t into a textual prompt and encode the prompt as language tokens E_{text}. Here, using a textual prompt to represent the point goal allows us to co-train the navigation data with open-world VQA data, which has been widely proven to mitigate the sim-to-real gap [59, 78]. We then feed the concatenation of the visual tokens E_{visual} and language tokens E_{text} into a large language model (implemented using Qwen2 [68]) to obtain the predicted action token E_{action}. Finally, the action token is processed by an action head (implemented with a two-layer MLP) to predict the velocity ($v_t = [v_t^{\text{x}}, v_t^{\text{y}}, v_t^{\text{yaw}}]$) of the robot.

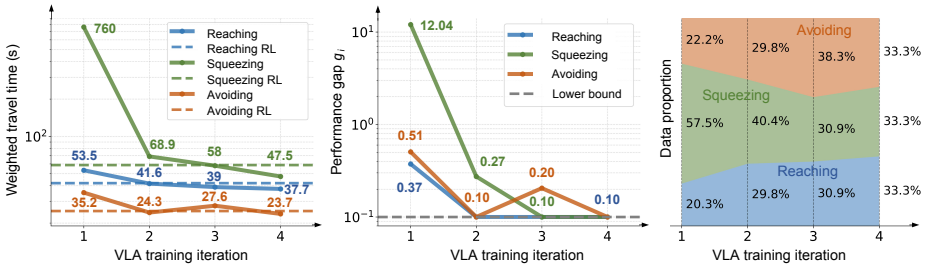
Here, we apply mean-square-error loss $L_{\text{action}} = \text{MSE}(v^{\text{Pred.}}, v^{\text{GT}})$ for action prediction, and retain the cross-entropy loss L_{QA} for open-world question-answering data, similar to [59]. The overall loss is defined as: $L = \beta L_{\text{action}} + L_{\text{QA}}$, where the β is set to 5 to balance the magnitude of the two loss terms. Notably, through a tiered architecture that incorporates visual tokens, our method achieves an inference speed of 7 Hz using a 7B-parameter LLM, while maintaining strong performance in challenging scenarios. This framework could be further improved by integrating acceleration techniques such as quantization [11, 25, 27].

3.4 RL Experts–VLA iteration

To distill expertise from multiple RL experts into the student VLA model, we design a two-stage training process.

Initial expert data collection and VLA finetuning. To endow the VLA model with initial navigation capability, we first collect a diverse set of trajectories generated by RL experts in simulation. For each expert, we run 64 parallel robots in their respective simulation environments, recording the observations, goals, and corresponding expert actions. Only trajectories that successfully reach the goal are retained, ensuring that erroneous actions from imperfect RL experts are excluded. All retained trajectories are aggregated into a dataset containing 500k steps, each represented as $[O_i, g_i, a_i]$, where a_i is the ground truth velocity. During collection, we randomized the height of the camera within $[0.4\,\text{m}, 0.5\,\text{m}]$ and the field of view (FOV) of each camera within $[100°, 140°]$. The initial dataset also contains 100k real-world VQA data. As shown in Figure 7, after being trained on the large-scale dataset, the initial VLA model performs well in the reaching scene but poorly in others, and in all cases underperforms the corresponding RL expert.

Teachers-student online training iteration. We collect online expert data iteratively in a data aggregation manner. The preliminarily trained VLA model is deployed in the three simulation scenes to collect actions from the corresponding RL teachers. However, the varying complexities of the three capabilities result in



(a) Weighted travel time across training iterations.

(b) Performance gaps across training iterations.

(c) Data proportion across training iterations.

Fig. 4: Online training dynamics across iterations. (a) Weighted travel time (WTT) of the VLA model compared with RL experts. (b) Performance gap g_i between the VLA model and RL experts. (c) Data proportion of the online collected data for reaching, squeezing, and avoiding. With four iterations, shrinking performance gaps lead the capability-balancing module to equalize sampling ratios, and VLA eventually matches or surpasses all experts in WTT, yielding a uniform data ratio.

uneven performance of VLA across the three scenes, like performing well in the reaching task but underperforming in the more challenging squeezing and avoiding tasks. This motivates the need for a more balanced use of training data from different scenes. To address this issue, we propose a capability-balanced data aggregation method. It leverages weighted travel time (WTT) W (the average time of successful episodes divided by success rate) to measure the performance gap between the VLA model and the RL experts. The gap is defined as:

$$g_{\text{Cap.}} = \max\left\{0, \frac{W^{\text{VLA}}_{\text{Cap.}} - W^{\text{RL}}_{\text{Cap.}}}{W^{\text{RL}}_{\text{Cap.}}}\right\} + \epsilon, \tag{5}$$

where $W^{\text{RL}}_{\text{Cap.}}$ represents the WTT of the reaching, squeezing, and avoiding experts, respectively, and $W^{\text{VLA}}_{\text{Cap.}}$ represents the WTT of the current VLA model in each scene. We set ϵ to 0.1 to maintain a set of data when the VLA model already outperforms the RL expert. Based on this gap, we compute the data proportion $p_{\text{Cap.}}$ for each capability as:

$$p_{\text{Cap.}} = \frac{g^{\alpha}_{\text{Cap.}}}{\sum_{\text{Cap.}} g^{\alpha}_{\text{Cap.}}}, \tag{6}$$

with $\alpha = 0.3$ to smooth the distribution. Intuitively, larger performance gaps lead to higher proportions of the corresponding expert data being aggregated, as illustrated in Figure 4. Finally, using the computed proportions $p_{\text{Cap.}}$, we aggregate the expert datasets in a capability-balanced manner and fine-tune the VLA. This enables the transfer of additional collision-avoidance and navigation skills. After fine-tuning, the VLA is re-evaluated and the process is repeated iteratively with updated $p_{\text{Cap.}}$ values until no further improvement is observed.

3.5 Implementation Details

RL Training strategy. Each RL expert is trained in IsaacLab [36] on an NVIDIA RTX 4090 GPU for 8-12 hours, using $N = 128$ parallel environments.

The policy adopts a history-aware actor-critic architecture. Both the actor and critic are implemented as three-layer MLPs with hidden dimensions of [512, 256, 128] and use the ELU activation function [7]. Regarding the sensory input, the depth values from the cameras are clipped to [0.01m, 4.0m] to filter out noise. The action distribution is initialized with a noise standard deviation of 0.2.

VLA training strategy. Our initial VLA model is fine-tuned on 8 NVIDIA H100 GPUs for about 5 hours, totaling 40 GPU hours. We leverage the pre-trained weights of both visual encoders (SigLIP [76]) and LLM (Qwen2-7B [68]). The initial training contains 500k steps from three RL experts and 100k visual question answering (VQA) data. For VQA data, we use a subset of the dataset in [51] and the frames are sampled at 1 FPS, following [79]. Each iteration of teacher-student training costs about 2 hours and contains 200k steps of online collected expert data and 40k VQA data.

Deployment strategy. We deploy our method on the Unitree GO2 robot. The VLA model is executed on a server equipped with an NVIDIA RTX 5090 GPU, while the onboard computer on the robot continuously sends requests and receives velocity commands. We use four fisheye cameras mounted on the front, right, back, and left sides of the robot to obtain the four-view real-time images, which are then undistorted into normal perspective images and fed into the VLA model. Given the velocity v_t inferred by the VLA model, the low-level controller follows the velocity until the next response. The average response fps is about 7 Hz, which is sufficient for our navigation task.

4 Experiments

4.1 Experiment Setup

Simulation environment setup. For comparative evaluation, we first assess MM-Nav on the public InternVLA-N1 System-1 (S1) point-goal navigation benchmark [22]. Following the benchmark protocol, we evaluate on 40 scenes from InternScenes [83], using the same 100 start-goal pairs as InternVLA-N1 and the Dingo robot platform with omnidirectional control. To further evaluate our method in more challenging settings, we additionally construct four custom test environments in IsaacLab [36]. Three fixed capability-specific scenes are designed to separately assess the three target capabilities, namely reaching, squeezing, and avoiding. In addition, we build a `Mixed` scene that combines static obstacles, dynamic obstacles, and narrow passages, all of which lie outside the VLA model's training distribution. All objects in the `Mixed` scene are rendered with materials visually distinct from those used during training, and the difficulty of the squeezing segment is intentionally reduced to keep the overall evaluation balanced. Each episode terminates when the robot reaches the goal, collides with an obstacle, or exceeds the time limit. The timeout is set to 90 seconds for the three capability-specific scenes and extended to 120 seconds for the more complex `Mixed` scene. Each method was evaluated over 100 episodes per scene following its original configuration.

Real-world environment setup. We conduct extensive real-world deployments across diverse indoor and outdoor environments to evaluate its sim-to-

Table 1: Results in InternVLA-N1 [22] System-1 point-goal navigation benchmark. For fair comparison with prior front-view baselines, we report a single-view variant.

Methods	Obs.	Home		Commercial	
		SR↑	SPL↑	SR↑	SPL↑
DD-PPO [62]	RGB-D	0.4	0.4	5.3	5.2
iPlanner [69]	Depth	43.0	40.6	54.6	52.8
ViPlanner [44]	RGB-D	45.0	43.2	63.7	61.9
LoGoPlanner [38]	RGB-D	57.3	52.4	67.1	63.9
InternVLA-N1(S1) [22]	RGB-D	60.0	55.6	71.4	68.2
NavDP [2]	RGB-D	60.3	54.7	74.1	70.5
SIDP [81]	RGB-D	63.2	56.5	81.2	73.4
Mixed-RL	Depth	22.0	10.4	26.9	22.8
Ours (without VQA)	RGB	76.4	73.7	73.9	72.6
Ours (single view)	RGB	79.9	77.6	86.7	84.7
Ours (multi-view)	RGB	**86.3**	**81.1**	**89.9**	**85.5**

real transfer and generalization ability under varied layouts, objects, and lights. From these trials, we select four representative scenarios for detailed analysis. The first two scenarios are shown in Figure 1: *Thin Wire Avoidance*, a corridor scene densely populated with many suspended thin wires, and *Cluttered Corridor*, a narrow office corridor filled with obstacles and partially transparent glass walls. The other two scenarios are shown in Figure 5: *Pedestrian Avoidance*, which involves pedestrians moving across and near the robot's forward path, and *Outdoor Alley*, a nighttime outdoor scene containing previously unseen obstacles such as a stone bollard and a bicycle. These examples are chosen to show how the capabilities learned in simulation transfer to real-world navigation.

Metrics and baselines. On the public InternVLA-N1 System-1 benchmark, we report two standard metrics: Success Rate (SR) and Success weighted by Path Length (SPL). On our custom IsaacLab benchmark, we report three metrics: (1) Success Rate (SR), (2) Collision Rate (CR), and (3) Weighted Travel Time (WTT), where WTT is defined as the average completion time of successful episodes divided by the SR. For the IsaacLab evaluation, each method is rolled out for 100 episodes in each scene. We compare MM-Nav against DD-PPO, iPlanner, ViPlanner, LoGoPlanner, InternVLA-N1(S1), NavDP, and SIDP on the public benchmark, and against iPlanner, ViPlanner, and NavDP in the custom IsaacLab environments.

4.2 Quantitative Results

The results of MM-Nav compared to other baselines on the InternVLA-N1 S1 benchmark are shown in Table 1. Our method achieves the highest performance across both Home and Commercial splits, significantly outperforming prior RGB-D and depth-based approaches, despite relying solely on RGB observations. In particular, MM-Nav improves the SR by approximately 25% over the strongest baseline. However, our RL expert itself performs poorly on this benchmark, suffering from severe domain gap in different simulation environments. These

Table 2: Quantitative comparison on IsaacLab simulator in three capability-specific scenes and a mixed scene. Each method was evaluated over 100 episodes per scene.

Methods	Reaching			Squeezing			Avoiding			Mixed		
	SR↑	CR↓	WTT↓	SR↑	CR↓	WTT↓	SR↑	CR↓	WTT↓	SR↑	CR↓	WTT↓
iPlanner [69]	19	81	93.4	2	94	881.25	15	85	73.4	4	94	736.9
ViPlanner [44]	43	57	42.2	4	96	452.8	22	78	36.36	18	82	215.4
NavDP [2]	69	31	**27.3**	18	82	115.4	27	73	30.0	23	77	178.6
Ours	**80**	**20**	31.0	**71**	**19**	**42.2**	**68**	**32**	**20.9**	**47**	**26**	**127.5**

results demonstrate that our multi-capability training paradigm effectively enhances generalization across diverse environments.

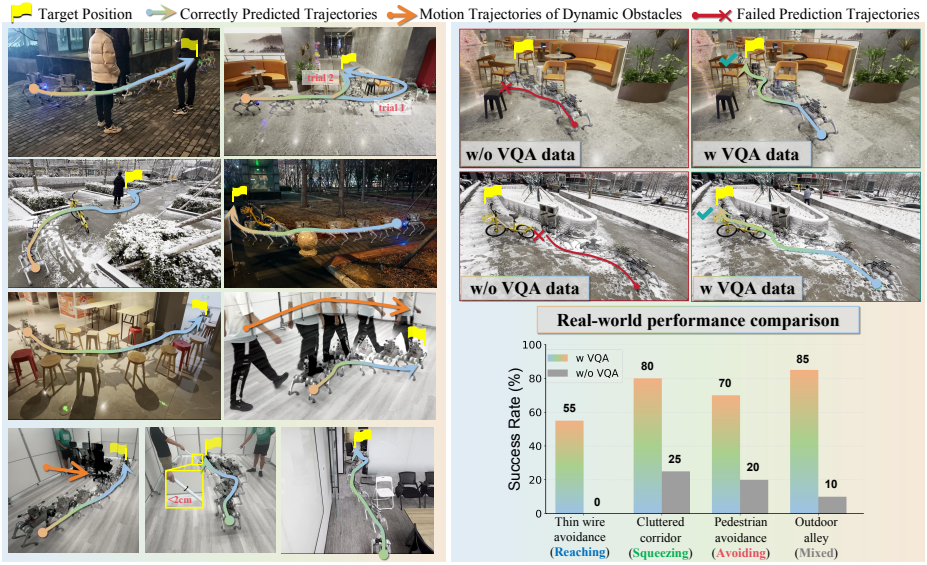
The simulation benchmark results on our custom IsaacLab benchmark are summarized in Table 2. Overall, MM-Nav achieves the highest SR, the lowest CR, and competitive or shortest WTT across almost all scenes, indicating both safe and efficient navigation. Among the baselines, NavDP [2] performs competitively in the Reaching scene but is fundamentally constrained by its single front-facing camera, which limits its field of view (FOV). This restriction leads to characteristic failures in complex scenarios, particularly when obstacles may approach from lateral or rear directions or remain partially occluded. ViPlanner [44] and iPlanner [69] suffer from similar FOV limitations and exhibit less responsive control behaviors, resulting in lower performance. In contrast, our 360° surround-view perception combined with continuous velocity control enables more robust behavior in cluttered, dynamic and constrained environments. The visualization results in simulators are provided in the supplementary material.

4.3 Qualitative Results

Across the four selected real-world scenarios, MM-Nav demonstrates strong zero-shot sim-to-real transfer. In *Thin Wire Avoidance*, the robot successfully avoids these thin obstacles, which are challenging for LiDAR-based systems to reliably perceive [61], indicating that the learned visual representation captures subtle obstacle cues and supports the *reaching* capability learned in simulation. In *Cluttered Corridor*, the robot navigates through the cluttered passage while handling unseen obstacles and challenging materials like glass, reflecting the *squeezing* capability learned in simulation. In *Pedestrian Avoidance*, the robot adjusts its actions online according to the pedestrians' trajectories and precisely avoids close-range interactions while maintaining progress toward the goal, demonstrating the learned *avoiding* capability. In *Outdoor Alley*, the robot accurately avoids these previously unseen obstacles under low-light conditions, further demonstrating robust real-world generalization.

4.4 Ablation Study

Generalization ability gained from VQA data. We conduct an ablation study on co-training VQA data to prove that large-scale real world VQA data is essential in helping model generalize to different simulation environments and bridge sim-to-real gap. The comparison result in unseen simulation environments is shown in Table 1, the VLA model trained without VQA data



(a) Real-world instances. (b) Performance ablation w/ vs. w/o QA data.

Fig. 5: Real-world Experiments. (a): Representative deployments of MM-Nav in diverse real-world environments under varying scene layouts and lighting conditions. (b): A real-world ablation study on VQA co-training.

suffers a large performance drop of more than 20% in both Home and Commercial scenes. Moreover, we evaluate the SR of MM-Nav with and without VQA data across several challenging real-world scenarios mentioned in Section 4.1. Each model is evaluated for 20 trials in every scenario. As shown in Figure 5 (b), the lack of VQA data leads to a very poor performance, unable to correctly avoid obstacles with thin or irregular shape. While the VLA model co-trained with VQA data demonstrates strong sim-to-real transfer ability.

Beyond the above performance ablations, we further analyze how VQA co-training affects navigation action prediction by visualizing the final-layer VLM hidden states used by the action head. Specifically, we feed synthetic and real-world observations into MM-Nav variants trained with 0%, 50%, and 100% VQA data, and analyze their action-relevant VLM hidden states using UMAP and gap-axis projection in Figure 6. Without VQA co-training, the simulation and real-world latents form clearly separated clusters, indicating a large sim-to-real visual gap. As the amount of VQA data increases, the two domains become progressively better aligned, with the gap-axis distance decreasing from 1.86 to 0.38. These results

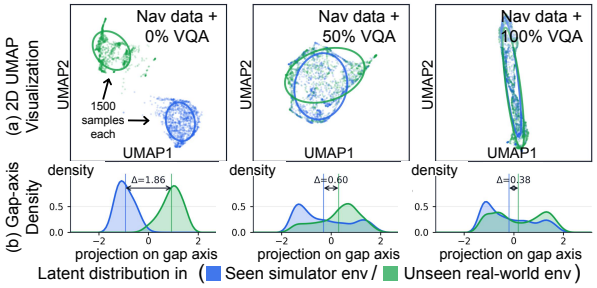


Fig. 6: Latent distributions under different VQA ratios.

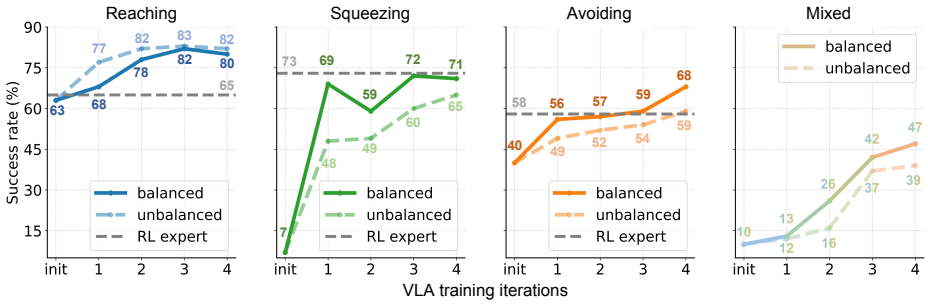


Fig. 7: Success rate of the VLA model after each iteration.

suggest that VQA co-training exposes the shared VLM backbone to real-image statistics, including lighting variations and complex scene structures, thereby helping align real observations with the simulation-trained latent space.

Multi-view vs. single-view observations. We further study the impact of our 360° surround-view input by training a single-view variant that uses only the front-facing RGB stream, while keeping the model, training data, and optimization settings unchanged. On the InternVLA-N1 S1 benchmark, multi-view consistently improves performance over single-view, yielding an average gain of +4.8 SR and +2.2 SPL across the Home and Commercial splits (Table 1). This validates that the additional side and rear views provide stronger spatial awareness beyond front-view-only policies, mitigating failures caused by limited FOV under clutter, occlusions, and tight passages.

Performance gains across online training iterations. The performance of the initial VLA model and its variants after the first four training iterations is evaluated in simulation, as shown in Figure 7. After the initial behavior-cloning training, the VLA model shows clear performance gaps across all three capabilities, particularly in squeezing. In the first iteration, the capability-balanced data aggregation method places more emphasis on squeezing, leading to substantial gains. As a result, the VLA model after the first iteration becomes comparable to, or even surpasses, the RL experts. In the second and third iterations, the reaching capability is further improved, although the VLA model had already exceeded the corresponding RL expert. This may be attributed to better integration of the squeezing and avoiding capabilities, which also benefits the relatively simpler reaching task. After the fourth iteration, performance across the three tasks converges, and no further iterations are conducted.

Capability-balanced data aggregation method. Comparison experiments are conducted to evaluate the effectiveness of our capability-balanced data aggregation method. Starting from the same initial VLA model, we train one set of iterations with balanced data and another with unbalanced data (Figure 7), while keeping the total number of samples per iteration the same. The results show that the capability-balanced method can complement underdeveloped capabilities in time, leading to faster and more stable training. Although the unbalanced setting achieves better performance on the reaching task, it fails to efficiently improve squeezing and avoiding. Overall, the capability-balanced method better

integrates data from different RL experts and prevents the VLA model from overlooking specific capabilities.

Experts Combination strategy. We investigate how combining three capability-specific RL teachers improves the VLA student. We train three VLA variants with the same total data budget, each using trajectories from one expert, and also train a single RL expert in the mixed scene requiring all three capabilities.

As shown in Table 3, the mixed-scene RL expert covers all capabilities but cannot match any specialized expert, leading to compromised performance. It also performs poorly on the InternVLA-N1 System-1 point-goal navigation benchmark (Table 2), further indicating the weak generalization ability of RL policies beyond their training distribution. Likewise, single-expert VLA variants perform well in-domain but generalize poorly to unseen capabilities. For example, training only on squeezing data may bias the model toward static obstacle interactions, while weakening its ability to actively avoid dynamic obstacles. In contrast, the mixed-data VLA model achieves stronger cross-capability performance, suggesting that different capabilities are complementary and help the student learn more transferable representations.

Table 3: Comparison of Individual Navigation Capability. We report the SR (100 episodes) of VLA models and RL experts trained on either single or mixed datasets. The best overall result is highlighted in **bold**, and the best result within each category is indicated with underline.

Methods	Reaching	Squeezing	Avoiding
Reaching-RL	65	2	33
Squeezing-RL	30	73	19
Avoiding-RL	59	0	58
Mixed-RL	45	58	42
Reaching-VLA	70	4	35
Squeezing-VLA	44	64	28
Avoiding-VLA	63	0	62
Mixed-VLA	**80**	**71**	**68**

5 Conclusion

We have presented MM-Nav, a multi-view VLA model that acquires robust visual navigation skills from a collective of specialized RL experts. Our training process consists of two key stages: first, an initial VLA finetuning phase where a student policy learns from a large offline dataset collected from the RL teachers; second, an online teachers-student training iteration where the student is deployed in simulation to receive on-the-fly, capability-balanced supervision for further refinement. MM-Nav demonstrates strong sim-to-real transfer and ultimately outperforms its RL experts, proving the synergistic benefit of learning multiple capabilities. MM-Nav provides a scalable and effective blueprint for training a new generation of general-purpose visual navigation agents. Future work includes investigating the cross-embodiment potential of our training strategy and further advancing visual-only navigation.

Acknowledgements

This work was supported by institutional research resources and computing platforms from Peking University and Galbot. The authors are also grateful to the members of PKU-EPIC and Galbot for helpful discussions, infrastructure support, and assistance with real-world robot deployment.

References

1. Cai, W., Huang, S., Cheng, G., Long, Y., Gao, P., Sun, C., Dong, H.: 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)
2. Cai, W., Peng, J., Yang, Y., Zhang, Y., Wei, M., Wang, H., Chen, Y., Wang, T., Pang, J.: Navdp: Learning sim-to-real navigation diffusion policy with privileged information guidance. arXiv preprint arXiv:2505.08712 (2025)
3. Castro, M.G., Rajagopal, S., Gorbatov, D., Schmittle, M., Baijal, R., Zhang, O., Scalise, R., Talia, S., Romig, E., de Melo, C., Boots, B., Gupta, A.: Vamos: A hierarchical vision-language-action model for capability-modulated and steerable navigation (2025)
4. Cheng, A.C., Ji, Y., Yang, Z., Gongye, Z., Zou, X., Kautz, J., Biyik, E., Yin, H., Liu, S., Wang, X.: Navila: Legged robot vision-language-action model for navigation. In: Proceedings of Robotics: Science and Systems. Los Angeles, CA, USA (Jun 2025)
5. Chhablani, G., Ye, X., Irshad, M.Z., Kira, Z.: Embodiedsplat: Personalized real-to-sim-to-real navigation with gaussian splats from a mobile device. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 25431–25441 (2025)
6. Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., Stoica, I., Xing, E.P.: Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality (March 2023), `https://lmsys.org/blog/2023-03-30-vicuna/`
7. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (elus). arXiv preprint arXiv:1511.07289 **4**(5), 11 (2015)
8. Cui, X., Liu, Q., Liu, Z., Wang, H.: Frontier-enhanced topological memory with improved exploration awareness for embodied visual navigation. In: European Conference on Computer Vision. pp. 296–313. Springer (2024)
9. Eftekhar, A., Weihs, L., Hendrix, R., Caglar, E., Salvador, J., Herrasti, A., Han, W., VanderBil, E., Kembhavi, A., Farhadi, A., et al.: The one ring: a robotic indoor navigation generalist. arXiv preprint arXiv:2412.14401 (2024)
10. Ehsani, K., Gupta, T., Hendrix, R., Salvador, J., Weihs, L., Zeng, K.H., Singh, K.P., Kim, Y., Han, W., Herrasti, A., et al.: Spoc: Imitating shortest paths in simulation enables effective navigation and manipulation in the real world. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16238–16250 (2024)
11. Frantar, E., Ashkboos, S., Hoefler, T., Alistarh, D.: Gptq: Accurate post-training quantization for generative pre-trained transformers. arXiv preprint arXiv:2210.17323 (2022)
12. He, H., Ma, Y., Squicciarini, B., Wu, W., Zhou, B.: From seeing to experiencing: Scaling navigation foundation models with reinforcement learning. In: International Conference on Learning Representations (2026), `https://openreview.net/forum?id=0c7nAZjyr5`, poster
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
14. He, T., Zhang, C., Xiao, W., He, G., Liu, C., Shi, G.: Agile but safe: Learning collision-free high-speed legged locomotion. arXiv preprint arXiv:2401.17583 (2024)

15. Hirose, N., Glossop, C., Shah, D., Levine, S.: Omnivla: An omni-modal vision-language-action model for robot navigation. arXiv preprint arXiv:2509.19480 (2025), https://arxiv.org/abs/2509.19480
16. Hirose, N., Glossop, C., Sridhar, A., Mees, O., Levine, S.: Lelan: Learning a language-conditioned navigation policy from in-the-wild video. In: Agrawal, P., Kroemer, O., Burgard, W. (eds.) Proceedings of The 8th Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 270, pp. 666–688. PMLR (06–09 Nov 2025), https://proceedings.mlr.press/v270/hirose25b.html
17. Hirose, N., Ignatova, L., Stachowicz, K., Glossop, C., Levine, S., Shah, D.: Learning to drive anywhere with model-based reannotation. arXiv preprint arXiv:2505.05592 (2025), https://arxiv.org/abs/2505.05592
18. Hirose, N., Shah, D., Sridhar, A., Levine, S.: Sacson: Scalable autonomous control for social navigation. IEEE Robotics and Automation Letters **9**(1), 49–56 (2023)
19. Hirose, N., Xia, F., Mart'ın-Mart'ın, R., Sadeghian, A., Savarese, S.: Deep visual mpc-policy learning for navigation. IEEE Robotics and Automation Letters **4**(4), 3184–3191 (2019)
20. Huang, C., Mees, O., Zeng, A., Burgard, W.: Visual language maps for robot navigation. arXiv preprint arXiv:2210.05714 (2022)
21. Huang, Z., Zhang, Y., Liu, J., Song, R., Tang, C., Ma, J.: Tic-vla: A think-in-control vision-language-action model for robot navigation in dynamic environments (2026)
22. Intern Robotics: Internvla-n1: An open dual-system vision-language navigation foundation model with learned latent plans (2025)
23. Kim, J., Sim, J., Kim, W., Sycara, K., Nam, C.: Enhancing safety of foundation models for visual navigation through collision avoidance via repulsive estimation. arXiv preprint arXiv:2506.03834 (2025)
24. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., et al.: Openvla: An open-source vision-language-action model. arXiv preprint arXiv:2406.09246 (2024)
25. Lang, J., Guo, Z., Huang, S.: A comprehensive study on quantization techniques for large language models. In: 2024 4th International Conference on Artificial Intelligence, Robotics, and Communication (ICAIRC). pp. 224–231. IEEE (2024)
26. Lee, J., Heo, J., Lee, G., Jun, H., Oh, J., Oh, S.: Rvn-bench: A benchmark for reactive visual navigation (2026)
27. Lin, J., Tang, J., Tang, H., Yang, S., Chen, W.M., Wang, W.C., Xiao, G., Dang, X., Gan, C., Han, S.: Awq: Activation-aware weight quantization for on-device llm compression and acceleration. Proceedings of machine learning and systems **6**, 87–100 (2024)
28. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. In: NeurIPS (2023)
29. Liu, W., Zhao, H., Li, C., Biswas, J., Pouya, S., Chang, Y.: Compass: Cross-embodiment mobility policy via residual rl and skill synthesis. arXiv preprint arXiv:2502.16372 (2025)
30. Liu, X., Li, J., Jiang, Y., Sujay, N., Yang, Z., Zhang, J., Abanes, J., Zhang, J., Feng, C.: Citywalker: Learning embodied urban navigation from web-scale videos. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 6875–6885 (2025)
31. Long, Y., Cai, W., Wang, H., Zhan, G., Dong, H.: Instructnav: Zero-shot system for generic instruction navigation in unexplored environment. arXiv preprint arXiv:2406.04882 (2024)
32. Long, Y., Li, X., Cai, W., Dong, H.: Discuss before moving: Visual language navigation via multi-expert discussions. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 17380–17387. IEEE (2024)

33. Lu, R., Meng, J., Zheng, W.S.: Pret: Planning with directed fidelity trajectory for vision and language navigation. In: European Conference on Computer Vision. pp. 72–88. Springer (2024)
34. Meng, X., Ratliff, N., Xiang, Y., Fox, D.: Scaling local control to large-scale topological navigation. In: 2020 IEEE International Conference on Robotics and Automation (ICRA). pp. 672–678. IEEE (2020)
35. Meng, X., Yang, X., Jung, S., Ramos, F., Jujjavarapu, S.S., Paul, S., Fox, D.: Aim my robot: Precision local navigation to any object. IEEE Robotics and Automation Letters (2025)
36. Mittal, M., Yu, C., Yu, Q., Liu, J., Rudin, N., Hoeller, D., Yuan, J.L., Singh, R., Guo, Y., Mazhar, H., Mandlekar, A., Babich, B., State, G., Hutter, M., Garg, A.: Orbit: A unified simulation framework for interactive robot learning environments. IEEE Robotics and Automation Letters **8**(6), 3740–3747 (2023). https://doi.org/10.1109/LRA.2023.3270034
37. Pan, B., Harley, A.W., Engelmann, F., Liu, C.K., Guibas, L.J.: Lookout: Real-world humanoid egocentric navigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 24977–24988 (2025)
38. Peng, J., Cai, W., Yang, Y., Wang, T., Shen, Y., Pang, J.: Logoplanner: Localization grounded navigation policy with metric-aware visual geometry. arXiv preprint arXiv:2512.19629 (2025)
39. Qi, Z., Zhang, W., Ding, Y., Dong, R., Yu, X., Li, J., Xu, L., Li, B., He, X., Fan, G., et al.: Sofar: Language-grounded orientation bridges spatial reasoning and object manipulation. arXiv preprint arXiv:2502.13143 (2025)
40. Qiao, Y., Liu, Q., Liu, J., Liu, J., Wu, Q.: Llm as copilot for coarse-grained vision-and-language navigation. In: European Conference on Computer Vision. pp. 459–476. Springer (2024)
41. Qin, L., Wang, M., Li, P., Zhou, W., Li, H.: Active perception meets rule-guided rl: A two-phase approach for precise object navigation in complex environments. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 7603–7612 (October 2025), https://openaccess.thecvf.com/content/ICCV2025/papers/Qin_Active_Perception_Meets_Rule-Guided_RL_A_Two-Phase_Approach_for_Precise_ICCV_2025_paper.pdf
42. Ross, S., Gordon, G., Bagnell, D.: 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)
43. Roth, P., Frey, J., Cadena, C., Hutter, M.: Learned perceptive forward dynamics model for safe and platform-aware robotic navigation. In: Robotics: Science and Systems (RSS) (2025)
44. Roth, P., Nubert, J., Yang, F., Mittal, M., Hutter, M.: Viplanner: Visual semantic imperative learning for local navigation. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 5243–5249. IEEE (2024)
45. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017)
46. Seneviratne, G., An, J., Ellahy, S., Weerakoon, K., Elnoor, M.B., Kannan, J.D., Sunil, A.T., Manocha, D.: Halo: Human preference aligned offline reward learning for robot navigation. In: Conference on Robot Learning (CoRL) (2025)
47. Shah, D., Osinski, B., Ichter, B., Levine, S.: Robotic navigation with large pre-trained models of language. Vision, and Action (2022)

48. Shah, D., Osiński, B., Levine, S., et al.: Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In: Conference on robot learning. pp. 492–504. PMLR (2023)
49. Shah, D., Sridhar, A., Bhorkar, A., Hirose, N., Levine, S.: Gnm: A general navigation model to drive any robot. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 7226–7233. IEEE (2023)
50. Shah, D., Sridhar, A., Dashora, N., Stachowicz, K., Black, K., Hirose, N., Levine, S.: Vint: A foundation model for visual navigation. arXiv preprint arXiv:2306.14846 (2023)
51. Shen, X., Xiong, Y., Zhao, C., Wu, L., Chen, J., Zhu, C., Liu, Z., Xiao, F., Varadarajan, B., Bordes, F., et al.: Longvu: Spatiotemporal adaptive compression for long video-language understanding. arXiv preprint arXiv:2410.17434 (2024)
52. Shi, X., Li, Z., Qiao, Y., Wu, Q.: Fast-smartway: Panoramic-free end-to-end zero-shot vision-and-language navigation (2025)
53. Song, C.H., Wu, J., Washington, C., Sadler, B.M., Chao, W.L., Su, Y.: Llm-planner: Few-shot grounded planning for embodied agents with large language models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2998–3009 (2023)
54. Sridhar, A., Shah, D., Glossop, C., Levine, S.: Nomad: Goal masked diffusion policies for navigation and exploration. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 63–70. IEEE (2024)
55. Sun, J.Q., Xing, X., Weng, H., Yeum, C.M., Crowley, M.: View invariant learning for vision-language navigation in continuous environments. arXiv preprint arXiv:2507.08831 (2025)
56. Wang, H., Tan, A.H., Fung, A., Nejat, G.: X-nav: Learning end-to-end cross-embodiment navigation for mobile robots. arXiv preprint arXiv:2507.14731 (2025)
57. Wang, H., Chen, J., Huang, W., Ben, Q., Wang, T., Mi, B., Huang, T., Zhao, S., Chen, Y., Yang, S., et al.: Grutopia: Dream general robots in a city at scale. arXiv preprint arXiv:2407.10943 (2024)
58. Wang, L., Xia, X., Zhao, H., Wang, H., Wang, T., Chen, Y., Liu, C., Chen, Q., Pang, J.: Rethinking the embodied gap in vision-and-language navigation: A holistic study of physical and visual disparities. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9455–9465 (October 2025), https://openaccess.thecvf.com/content/ICCV2025/papers/Wang_Rethinking_the_Embodied_Gap_in_Vision-and-Language_Navigation_A_Holistic_Study_ICCV_2025_paper.pdf
59. Wang, S., Zhang, J., Li, M., Liu, J., Li, A., Wu, K., Zhong, F., Yu, J., Zhang, Z., Wang, H.: Trackvla: Embodied visual tracking in the wild. arXiv preprint arXiv:2505.23189 (2025)
60. Wang, T., Zhao, X., Cai, W., Sun, C.: Imaginenav++: Prompting vision-language models as embodied navigator through scene imagination (2026)
61. Wang, Z., Ma, T., Jia, Y., Yang, X., Zhou, J., Ouyang, W., Zhang, Q., Liang, J.: Omni-perception: Omnidirectional collision avoidance for legged locomotion in dynamic environments. arXiv preprint arXiv:2505.19214 (2025)
62. Wijmans, E., Kadian, A., Morcos, A., Lee, S., Essa, I., Parikh, D., Savva, M., Batra, D.: DD-PPO: Learning near-perfect pointgoal navigators from 2.5 billion frames. In: International Conference on Learning Representations (ICLR) (2020)
63. Xiao, W., Xue, H., Tao, T., Kalaria, D., Dolan, J.M., Shi, G.: Anycar to anywhere: Learning universal dynamics model for agile and adaptive mobility. In: 2025 IEEE International Conference on Robotics and Automation (ICRA). pp. 8819–8825. IEEE (2025)

64. Xu, P., Gong, X., Mu, Y.: Navq: Learning a q-model for foresighted vision-and-language navigation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6327–6341 (October 2025), https://openaccess.thecvf.com/content/ICCV2025/papers/Xu_NavQ_Learning_a_Q-Model_for_Foresighted_Vision-and-Language_Navigation_ICCV_2025_paper.pdf
65. Xu, X., Luo, S., Yang, Y., Li, Y.L., Lu, C.: Disco: Embodied navigation and interaction via differentiable scene semantics and dual-level control. In: European Conference on Computer Vision. pp. 108–125. Springer (2024)
66. Xu, Z., Han, X., Shen, H., Jin, H., Shimada, K.: Navrl: Learning safe flight in dynamic environments. IEEE Robotics and Automation Letters (2025)
67. Xue, X., Hu, J., Luo, M., Xie, S., Chen, J., Xie, Z., Quan, K., Guo, W., Xu, M., Chu, Z.: Omninav: A unified framework for prospective exploration and visual-language navigation (2026)
68. Yang, A., Yang, B., Hui, B., Zheng, B., Yu, B., Zhou, C., Li, C., Li, C., Liu, D., Huang, F., Dong, G., Wei, H., Lin, H., Tang, J., Wang, J., Yang, J., Tu, J., Zhang, J., Ma, J., Xu, J., Zhou, J., Bai, J., He, J., Lin, J., Dang, K., Lu, K., Chen, K., Yang, K., Li, M., Xue, M., Ni, N., Zhang, P., Wang, P., Peng, R., Men, R., Gao, R., Lin, R., Wang, S., Bai, S., Tan, S., Zhu, T., Li, T., Liu, T., Ge, W., Deng, X., Zhou, X., Ren, X., Zhang, X., Wei, X., Ren, X., Fan, Y., Yao, Y., Zhang, Y., Wan, Y., Chu, Y., Liu, Y., Cui, Z., Zhang, Z., Fan, Z.: Qwen2 technical report. arXiv preprint arXiv:2407.10671 (2024)
69. Yang, F., Wang, C., Cadena, C., Hutter, M.: iplanner: Imperative path planning. arXiv preprint arXiv:2302.11434 (2023)
70. Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J., Zhao, H.: Depth anything: Unleashing the power of large-scale unlabeled data. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10371–10381 (2024)
71. Yang, Y., Sun, J., Kou, S., Wang, Y., Deng, Z.: Lohovla: A unified vision-language-action model for long-horizon embodied tasks (2025)
72. Yao, J., Zhang, X., Xia, Y., Wang, Z., Roy-Chowdhury, A.K., Li, J.: Towards generalizable safety in crowd navigation via conformal uncertainty handling. arXiv preprint arXiv:2508.05634 (2025)
73. Yokoyama, N., Ha, S., Batra, D., Wang, J., Bucher, B.: Vlfm: Vision-language frontier maps for zero-shot semantic navigation. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 42–48. IEEE (2024)
74. Yu, Z., Long, Y., Yang, Z., Zeng, C., Fan, H., Zhang, J., Dong, H.: Correctnav: Self-correction flywheel empowers vision-language-action navigation model (2025)
75. Zeng, K.H., Zhang, Z., Ehsani, K., Hendrix, R., Salvador, J., Herrasti, A., Girshick, R., Kembhavi, A., Weihs, L.: Poliformer: Scaling on-policy rl with transformers results in masterful navigators. arXiv preprint arXiv:2406.20083 (2024)
76. Zhai, X., Mustafa, B., Kolesnikov, A., Beyer, L.: Sigmoid loss for language image pre-training. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 11975–11986 (2023)
77. Zhang, J., Li, A., Qi, Y., Li, M., Liu, J., Wang, S., Liu, H., Zhou, G., Wu, Y., Li, X., Fan, Y., Li, W., Chen, Z., Gao, F., Wu, Q., Zhang, Z., Wang, H.: Embodied navigation foundation model. arXiv preprint arXiv:2509.12129 (2025), https://arxiv.org/abs/2509.12129
78. Zhang, J., Wang, K., Wang, S., Li, M., Liu, H., Wei, S., Wang, Z., Zhang, Z., Wang, H.: Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks. Robotics: Science and Systems (2025)

79. Zhang, J., Wang, K., Xu, R., Zhou, G., Hong, Y., Fang, X., Wu, Q., Zhang, Z., Wang, H.: Navid: Video-based vlm plans the next step for vision-and-language navigation. Robotics: Science and Systems (2024)
80. Zhang, L., Liu, Y., Zhang, Z., Aghaei, M., Hu, Y., Gu, H., Alomrani, M.A., Bravo, D.G.A., Karimi, R., Hamidizadeh, A., Xu, H., Huang, G., Zhang, Z., Cao, T., Qiu, W., Quan, X., Hao, J., Zhuang, Y., Zhang, Y.: Mem2ego: Empowering vision-language models with global-to-ego memory for long-horizon embodied navigation (2025)
81. Zhang, R., Hou, J., Cheng, C., Chen, Q., Wang, T., Zhao, W.: Self-imitated diffusion policy for efficient and robust visual navigation. arXiv preprint arXiv:2601.22965 (2026)
82. Zhang, W., Liu, H., Qi, Z., Wang, Y., Yu, X., Zhang, J., Dong, R., He, J., Wang, H., Zhang, Z., et al.: Dreamvla: a vision-language-action model dreamed with comprehensive world knowledge. arXiv preprint arXiv:2507.04447 (2025)
83. Zhong, W., Cao, P., Jin, Y., Luo, L., Cai, W., Lin, J., Wang, H., Lyu, Z., Wang, T., Dai, B., et al.: Internscenes: A large-scale simulatable indoor scene dataset with realistic layouts. arXiv preprint arXiv:2509.10813 (2025)
84. Zhong, Y., Feng, C., Yan, F., Liu, F., Zheng, L., Ma, L.: Robotron-nav: A unified framework for embodied navigation integrating perception, planning, and prediction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6416–6425 (October 2025), https://openaccess.thecvf.com/content/ICCV2025/papers/Zhong_RoboTrom-Nav_A_Unified_Framework_for_Embodied_Navigation_Integrating_Perception_Planning_ICCV_2025_paper.pdf
85. Zhou, G., Hong, Y., Wang, Z., Wang, X.E., Wu, Q.: Navgpt-2: Unleashing navigational reasoning capability for large vision-language models. In: European Conference on Computer Vision. pp. 260–278. Springer (2025)
86. Zhou, G., Hong, Y., Wu, Q.: Navgpt: Explicit reasoning in vision-and-language navigation with large language models. arXiv preprint arXiv:2305.16986 (2023)
87. Zhu, D., Chen, J., Haydarov, K., Shen, X., Zhang, W., Elhoseiny, M.: Chatgpt asks, blip-2 answers: Automatic questioning towards enriched visual descriptions. arXiv preprint arXiv:2303.06594 (2023)
88. Zhu, Y., Yang, S.M., Magnusson, M., Wang, A.: Hicrowd: Hierarchical crowd flow alignment for dense human environments. arXiv preprint arXiv:2602.05608 (2026)