

NavDP: Learning Sim-to-Real Navigation Diffusion Policy with Privileged Information Guidance

Wenzhe Cai^{*,1}, Jiaqi Peng^{*,1,2}, Yuqiang Yang¹, Yujian Zhang³, Meng Wei^{1,4}
Hanqing Wang¹, Yilun Chen¹, Tai Wang^{1,†}, Jiangmiao Pang^{1,†}

Abstract—Learning to navigate in dynamic and complex open-world environments is a critical yet challenging capability for autonomous robots. Existing approaches often rely on cascaded modular frameworks, which require extensive hyperparameter tuning or learning from limited real-world demonstration data. In this paper, we propose Navigation Diffusion Policy (NavDP), an end-to-end network trained solely in simulation that enables zero-shot sim-to-real transfer across diverse environments and robot embodiments. The core of NavDP is a unified transformer-based architecture that jointly learns trajectory generation and trajectory evaluation, both conditioned solely on local RGB-D observations. By learning to predict critic values for contrastive trajectory samples, our proposed approach effectively leverages supervision from privileged information available in simulation, thereby fostering accurate spatial understanding and enabling the distinction between safe and dangerous behaviors. To support this, we develop an efficient data generation pipeline in simulation and construct a large-scale dataset encompassing over one million meters of navigation experience across 3,000 scenes. Empirical experiments in both simulated and real-world environments demonstrate that NavDP significantly outperforms prior state-of-the-art methods. Furthermore, we identify key factors influencing the generalization performance of NavDP. The dataset and code are publicly available at <https://wzca99.github.io/navigation-diffusion-policy.github.io>.

I. INTRODUCTION

Navigation in dynamic open worlds is a fundamental yet challenging skill for robots. For pursuing embodied intelligent generalists, the navigation system is expected to be capable of zero-shot generalization across different embodiments and unstructured scenes. However, the traditional modular-based methods suffer from system latency and compounding errors which limits their performance, while the scarcity of high-quality data limits the scale-up training and performance of learning-based methods. Although several studies have tried to address this problem by collecting robot trajectories in the real world [1], [2], [3], the scaling process is still time-consuming and expensive.

In contrast, simulation data is diverse and scalable. With large-scale 3D digital replica scenes available [4], [5], [6], [7], [8], [9], we can efficiently generate customized infinite navigation trajectories with different types of observations and goals. Furthermore, with the increasing diversity of 3D assets and rapid progress of neural rendering algorithms, the long-standing sim-to-real gap problem can also be alleviated. To learn generalized navigation policies, imitation learning

[10], [11] typically relies on positive demonstrations, but it lack interaction and fail to incorporate negative feedback from the environment. In contrast, reinforcement learning (RL)-based methods [12], [13] can learn from interactions with reward signals, but suffer from low data efficiency.

In this paper, we propose a novel end-to-end transformer-based learning framework to combine the advantages of these two streams, **Navigation Diffusion Policy (NavDP)**, which achieves zero-shot sim-to-real policy transfer and cross-embodiment generalization with only simulation data. Our proposed framework leverages the efficiency of imitation learning and the expressiveness of a diffusion process to model the multi-modal distribution of expert demonstrations. To enable counterfactual reasoning, we adapt the concept of the critic value function from reinforcement learning and train NavDP to predict state-action values for both positive and negative trajectories. Our framework can fully take advantage of the privileged information in the simulation from two aspects: On the one hand, the trajectory generation can be trained under the guidance from global-optimal planner within simulation environments. On the other hand, the critic function can learn from arbitrary generated contrastive action samples with the global Euclidean Signed Distance Field (ESDF) available in simulation as a fine-grained guidance. To support large-scale training of NavDP, we developed a highly efficient navigation data engine capable of generating 2,500 trajectories per GPU per day—achieving a 20× improvement in efficiency over real-world data collection. This enables the creation of a comprehensive dataset encompassing over one million meters of robot navigation experience across 3,000 diverse scenes. We conduct extensive empirical evaluations in both simulated and real-world environments. The results demonstrate that our proposed NavDP outperforms previous state-of-the-art approaches by a large margin.

II. RELATED WORKS

Robot Diffusion Policy. Advanced generative models have shown great potential in capturing the multimodal distribution of robot policy learning. The diffusion policy [14] was the first to introduce the diffusion process into manipulation tasks, sparking numerous efforts to enhance its capabilities. These enhancements span various aspects, including state representations [15], [16], [17], [18], inference speed [19], [20], and deployment across diverse robot applications [21], [22], [23], [24]. However, as diffusion policies operate within an offline imitation learning framework, achieving strong real-world performance often depends on real-world teleop-

¹Shanghai AI Lab, ²Tsinghua University, ³Zhejiang University ⁴The University of Hong Kong

*Equal Contribution, †Corresponding Authors

This work is supported by Shanghai Artificial Intelligence Laboratory

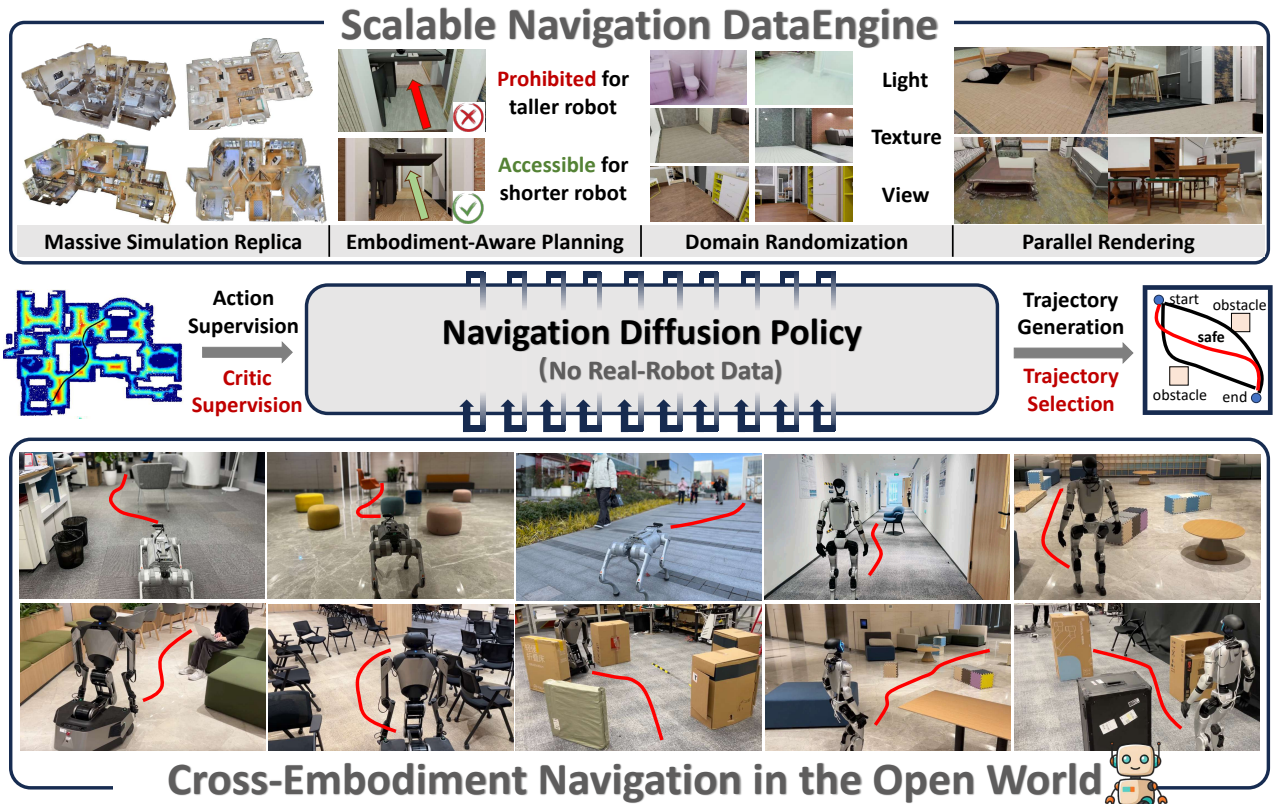


Fig. 1: NavDP is solely trained with simulation data but can achieve zero-shot sim-to-real transfer to different types of robots. By learning from the prioritized knowledge in the simulation data, NavDP forms commonsense spatial understanding ability for navigation task and can adaptively select safe navigation routes towards the goal without any maps.

eration datasets, which are labor-intensive and challenging to scale up. In contrast, our approach develops robot policies entirely from scalable simulation datasets. To enhance generalization and ensure safety during sim-to-real transfer, we introduce a critic function to estimate the safety of policy outputs. This mechanism leverages privileged simulation data to enable the diffusion policy to understand the consequences of actions, improving both safety and performance.

End-to-End Visual Navigation Models. Recent end-to-end visual navigation models have demonstrated significant potential in cross-embodiment adaptation and multi-task generalization [25], [26], [27], [28], [29], [30], [31], [32]. These approaches tackle navigation challenges at various levels of abstraction. Vision-Language-Action (VLA) models [30], [31], [32], [33] offer flexibility by leveraging language instructions for task specification. In contrast, end-to-end local navigation models excel in cross-embodiment generalization and demonstrate superior adaptability with real-time inference in open-world environments [25], [26], [11]. In this paper, we focus on developing efficient end-to-end cross-embodiment navigation system-1 which can seamlessly attach to the VLM for generalizable navigation task execution skills in dynamic open-world.

III. DATAENGINE

Robot Model. We build the robot as a cylindrical rigid body with a two-wheel differential drive model for cross-

embodiment generalizability. The navigation safe radius of the robots is set to $r_b = 0.25\text{m}$. To imitate the variation of observation views across different robot embodiments, we assume one RGB-D camera is installed on the top of the robot and the height of the robot h_b is randomized in the range $(0.25\text{m}, 1.25\text{m})$. Objects that are higher than the camera’s configuration height are not considered as obstacles during the navigation trajectory planning process. To ensure the local navigable area remains visible within the field of view, the camera’s pitch angle is randomized within the range $(-30^\circ, 0^\circ)$, depending on the robot’s height. We use two configurations to set the camera field of view, one follows RealSense D435i with horizontal field of view (HFOV) and vertical field of view (VFOV) set to $(69^\circ, 42^\circ)$, and the other follows Zed 2 with FOV set to $(110^\circ, 70^\circ)$.

Trajectory Generation. To generate collision-free robot navigation trajectories, we first convert the scene meshes into a voxel map with a voxel size of 0.05m to estimate the Euclidean Signed Distance Field (ESDF) of the navigable areas. Navigable areas are defined as voxel elements with z -axis coordinates below the threshold h_{nav} , while obstacle areas are defined as voxel elements with z -axis coordinates exceeding the threshold h_{obs} . The thresholds h_{nav} and h_{obs} vary across scenes and depend on the robot height h_b . Voxels with distance values lower than the robot radius r_b are truncated to prevent collisions. The ESDF map of the navigable area is downsampled to 0.2m resolution to

facilitate efficient A* path planning. Navigation start and target points are selected randomly from the navigable area, and the A* algorithm generates a planned path $\tau^* = [(x_0, y_0), (x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)]$. For each waypoint (x_n, y_n) , a greedy search is performed in a local area of the original ESDF map to refine the position by maximizing the distance to nearby obstacles. This refinement process shifts waypoints further from obstacles. Finally, the refined waypoints are smoothed into a continuous navigation trajectory using cubic spline interpolation.

Scene Assets and Render Engine. Following the pipeline described in the previous section, we can generate a large-scale dataset of robot navigation trajectories and corresponding RGB-D rendering results across diverse scenes. We use BlenderProc [34] to render photorealistic RGB and depth images along the navigation trajectories. We collect navigation trajectories from over 3,000 scenes selected from 3D-Front [6], HSSD [7], HM3D [8], Replica [4], Gibson [35], and Matterport3D [5]. For each scene, we sample 100 pairs of starting points and destinations. We adapt several domain randomization techniques to further improve the data diversity, which includes lighting condition randomization, view randomization as well as texture randomization. After data filtering, the final dataset comprises over 200K trajectories covering more than 1M meters. Compared with previous navigation dataset, our data dominates in diversity and collection efficiency as shown in Table I. The dataset will be open-sourced in the near future.

Dataset	Scene	Distance (Km)	Hour	Image	Collection
GoStanford [1]	27	25.5	16.7	178K	Teleop
RECON [36]	9	152.5	40	610K	Autonomous
SCAND [2]	1	40	8.7	100K	Teleop
SACSoN [3]	5	58	75	241K	Autonomous
AMR [10]	54	-	-	7.5M	Simulation
NavDP (ours)	3154	1627.1	452	40M	Simulation

TABLE I: Quantitative comparison of navigation datasets.

IV. NAVIGATION DIFFUSION POLICY

NavDP consists of a multi-modal encoder to fuse RGB and depth observations and a unified transformer-based network for both trajectory generation and critic value prediction. The NavDP network architecture is shown in Figure 2. The trajectory generation aims to plan M dense waypoints for robots to follow while the critic value prediction aims to predict scores corresponding to the safety of trajectories.

Multi-Modal Encoder. NavDP processes RGB-D images and a navigation goal as inputs. To incorporate historical information, we feed multi-frame RGB images of length N and use a pre-trained DepthAnything [37] encoder to extract 256 patch tokens from each RGB frame. To ensure alignment with absolute physical scale for trajectory generation, we introduce an additional ViT encoder—trained from scratch—to process a single frame of depth observation, which also produces 256 tokens. As the depth input may suffer from sim-to-real gap, here we only use the depth within range

(0.1m, 5m). To fuse the RGB-D inputs, we apply lightweight transformer decoder layers with learnable queries, compressing the original $(N + 1) \times 256$ tokens into $N \times 16$ compact tokens. The navigation goal follows the PointGoal task definition, where a 2-dimensional vector (x_g, y_g) represents the goal’s relative coordinates with respect to the current state. We employ MLP layers to encode the navigation goal and project it into the same dimensional space as the RGB-D tokens for subsequent processing. For no-goal tasks, we use a full-zero tensor as the goal embedding.

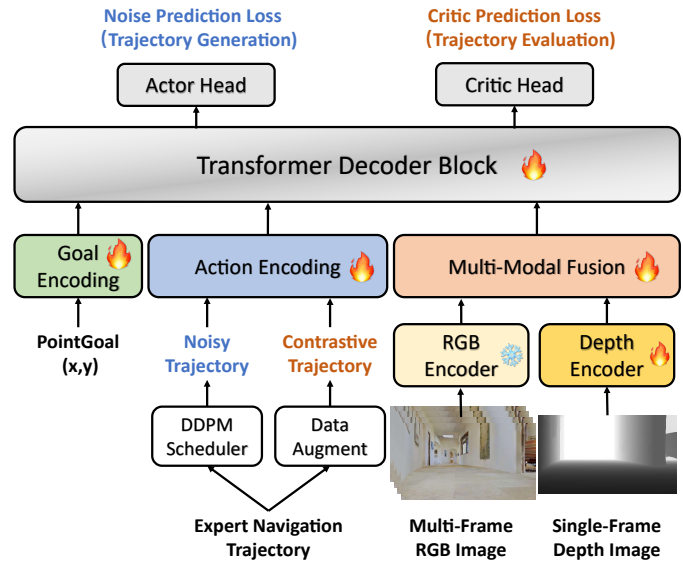


Fig. 2: Overview of the network architecture. NavDP is conditioned on RGB-D observations and navigation trajectories. During training, Gaussian noise is added to the ground-truth trajectory according to the DDPM scheduler, and the actor head is trained to predict the injected noise. Simultaneously, the ground-truth trajectories are augmented to create both collision-free and collision scenarios, and the critic head is trained to assign contrastive scores to these trajectories.

Unified Policy Transformer. We develop a simple yet effective transformer decoder-based architecture that supports both diffusion-based trajectory generation and trajectory evaluation. For trajectory generation, the objective is to predict the injected noise conditioned on a noisy trajectory. For trajectory evaluation, the goal is to predict a score conditioned on an arbitrary trajectory. To this end, we use an MLP-based action encoder to extract trajectory embeddings, which serve as queries in the cross-attention mechanism. The fused RGB-D tokens, along with a token representing the diffusion timestep, act as keys and values in the attention process. We employ multiple transformer decoder layers to process the input tokens and use two separate output heads for the two tasks. All network weights are shared between tasks; the distinction lies in the input queries and the attention masks applied to the keys and values. In the trajectory generation task, queries are extracted from noisy trajectories with noise added according to the DDPM [38] scheduler, and cross-attention attends to all keys

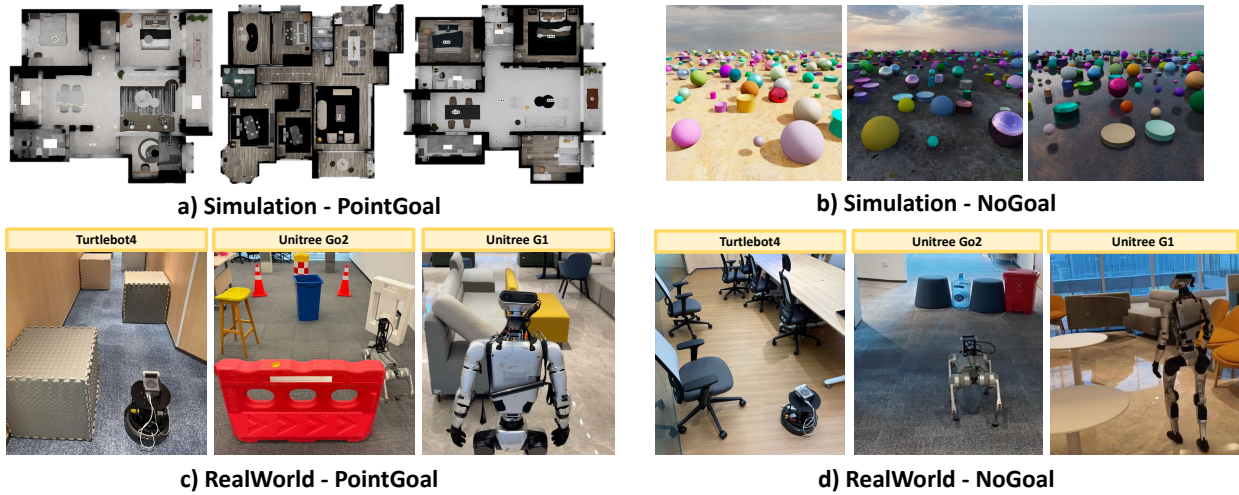


Fig. 3: An overview of the evaluation scenes, including both simulation and real-world. In simulation, we adapt 10 home scenes, 10 commercial scenes for point-goal task, and 10 scenes with clutter layout for no-goal task. In real-world, we evaluate different navigation policy on Turtlebot4, Unitree Go2, G1 and Galaxea R1 in both indoor and outdoor scenes.

and values. In the trajectory evaluation task, queries are derived from expert-demonstrated trajectories with random rotation augmentations, and the cross-attention excludes the timestep token. During inference, NavDP first generates a batch of candidate trajectories using the trajectory generation head and then selects the best trajectory via the evaluation head, thereby enabling safer task execution.

Training Details. The training objectives for the trajectory generation task are a weighted sum of mean squared error (MSE) losses of both predicted noises for point-goal and no-goal tasks. Denote k as the denoising steps, ϵ_k as the sampled noise following the DDPM scheduler at timestep k , d_t as the current depth image, $I_{t-N:t}$ as the RGB observations, g_t as the navigation goal, τ as the expert demonstration trajectories without noise, and \mathcal{D} as the entire dataset. The loss function for the actor head can be written as follows:

$$\mathcal{L}_{\text{act}}^{\text{ng}} = \mathbb{E}_{\tau, g, d, I \sim \mathcal{D}} [(\epsilon_k - \epsilon_{\theta}(\tau + \epsilon_{k+1}, k, d_t, I_{t-N:t}))^2] \quad (1)$$

$$\mathcal{L}_{\text{act}}^{\text{pg}} = \mathbb{E}_{\tau, g, d, I \sim \mathcal{D}} [(\epsilon_k - \epsilon_{\theta}(\tau + \epsilon_{k+1}, k, g_t, d_t, I_{t-N:t}))^2] \quad (2)$$

$$\mathcal{L}_{\text{act}} = \alpha \cdot \mathcal{L}_{\text{act}}^{\text{ng}} + \beta \cdot \mathcal{L}_{\text{act}}^{\text{pg}} \quad (3)$$

We set $\alpha = 0.5$ and $\beta = 0.5$ by default. ϵ_{θ} is the entire noise prediction network. For the trajectory evaluation task, we define the label critic value with respect to both the absolute ESDF value and difference of ESDF value along the trajectory waypoints. Concretely, denote the augmented expert demonstration trajectory as $\hat{\tau}$, the ESDF value at m -th waypoint on the augmented trajectory as $d_{\hat{\tau}}^m$, the label critic value is defined as follows:

$$V(\hat{\tau}) = \gamma \cdot \sum_{m=0}^M (d_{\hat{\tau}}^{m+1} - d_{\hat{\tau}}^m) + \lambda \cdot \frac{1}{M} \sum_{m=0}^M \mathbb{I}(d_{\hat{\tau}}^m < d_{\text{safe}}) \quad (4)$$

d_{safe} is a threshold representing the collision radius. Then, denote the score prediction network as V_{θ} , the loss function for critic head can be written as the follows:

$$\mathcal{L}_{\text{critic}} = \mathbb{E}_{I, d, \tau \sim \mathcal{D}} [V(\hat{\tau}) - V_{\theta}(I_{t-N:t}, D_t, \hat{\tau})] \quad (5)$$

The whole NavDP network is jointly trained in one stage with respect to the sum of \mathcal{L}_{act} and $\mathcal{L}_{\text{critic}}$. Additional hyper-parameters are shown in Table II.

Hyper-parameters	Value
Diffusion Step	10
Prediction Waypoints M	24
RGB History Size N	8
Safe Distance Threshold d_{safe}	0.5
Training GPU	32 A100 cards
Training Batchsize	2048
Learning Rate	1e-4
Learning Rate Decay	Linear
Training GPU Hours	24×32

TABLE II: Table of hyper-parameters of training.

V. EXPERIMENTS

A. Evaluation and Metrics

We evaluate our approach with point-goal navigation and no-goal exploration tasks across 4 different robots in both simulation and real-world. We build the simulation benchmark based on IsaacSim which offers high-fidelity physical simulation and use a wheeled robot - ClearPath Dingo as the navigator. For the point-goal navigation task, we collect 20 realistic scenes (10 home, 10 commercial) from GRUtopia [9], covering a wide range of scenarios including home, hospital, supermarket, etc. For the no-goal navigation task, we generate 10 challenging cluttered scenes with random obstacles as the for evaluation. We evaluate 2,000 episodes for the point-goal task and 1,000 episodes for the no-goal task.

In real-world benchmark, we evaluate the performance of Unitree Go2, Turtlebot4 and Unitree G1. For the point-goal navigation task, we set up 3 different indoor scenarios with challenging layout and evaluate 10 episodes for each scene with one embodiment. For the no-goal exploration

Point-Goal Nav	Dingo-Sim		Cross-Embodiment Real-world			
	SR(\uparrow)	SPL(\uparrow)	SR(\uparrow)	Turtlebot(\uparrow)	Unitree-Go2(\uparrow)	Unitree-G1(\uparrow)
DD-PPO [39]	8.6	8.5	-	-	-	-
EgoPlanner [40]	-	-	40.0	5/10	3/10	-
iPlanner [27]	54.1	51.2	16.7	0/10	5/10	0/10
ViPlanner [28]	60.9	58.6	53.3	5/10	4/10	7/10
NavDP (Ours)	67.2	62.6	76.7	9/10	7/10	7/10

TABLE III: Quantitative Evaluation Results of Point-Goal Navigation. We report both detailed success rate for each embodiment and the overall success rate in real-world experiments. Our proposed NavDP outperforms the previous state-of-the-art approach in simulation by **6.3%** success rate and **4.0%** SPL, **23.4%** success rate in real-world evaluation.

No-Goal Nav	Dingo-Sim		Cross-Embodiment Real-world			
	Time(\uparrow)	Area(\uparrow)	Time(\uparrow)	Turtlebot(\uparrow)	Unitree-Go2(\uparrow)	Unitree-G1(\uparrow)
GNM [25]	12.5	29.6	15.2	9.9	12.9	23.0
ViNT [26]	18.9	46.6	13.1	15.6	15.8	8.0
NoMad [11]	36.6	85.7	29.3	17.4	37.2	33.5
NavDP (Ours)	106.2	274.1	112.9	114.2	143.3	81.3

TABLE IV: Quantitative Evaluation Results of No-Goal Navigation. We report both detailed exploration time for each embodiment and the overall time. Our proposed NavDP achieves a nearly **2.9x** performance in exploration time and **3.1x** in exploration area better than the previous method in simulation, and **3.8x** performance in real-world exploration time.

task, we evaluate the cross-embodiment generalization in another 3 large indoor scenarios, including corridor, hall and meeting room. A brief visualization of the simulation and real-world evaluation scenarios are provided in Figure 3. In point-goal navigation task, we evaluate two metrics: **Success Rate (SR)** and **Success Weighted by Path Length (SPL)**, which measures task completion ratio and path efficiency. In no-goal navigation task, we evaluate another two metrics - **Time** and **Area**, which represents the the average time in seconds before one collision happens and the average exploration area. Both help evaluates the overall collision avoidance skill and planning consistency.

B. Experiment Analysis

In this section, we aim to address the following research questions through both quantitative and qualitative experimental results:

- **Q1:** How well does the proposed NavDP generalize across different robot platforms?
- **Q2:** What are the advantages of our method compared to baseline approaches?
- **Q3:** How well does our method generalize to in-the-wild indoor and outdoor environments?
- **Q4:** What are the key factors that influence the overall performance of the model?
- **Q5:** Is the domain randomization in navigation data essential for achieving cross-embodiment generalization?

For **Q1** and **Q2**, we compare our proposed NavDP with a range of baseline methods across both navigation tasks. The

baselines include learning-based approaches — GNM [25], ViNT [26], NoMaD [11], DD-PPO [39], iPlanner [27], and ViPlanner [28], as well as the planning-based approach EgoPlanner [40]. In the PointGoal navigation task, NavDP outperforms the previous state-of-the-art learning-based method by **6.3%** in Success Rate (SR) within simulation and achieves an average improvement of **23.0%** in cross-embodiment real-world experiments (as shown in Table III). In contrast, despite the baseline method DD-PPO has been extensively trained with reinforcement learning in the Habitat simulator, it demonstrates poor generalization to out-of-distribution scenarios with different action spaces and camera configurations. Compared with other approaches, our proposed NavDP offers three key advantages:

- **Temporal Consistency:** iPlanner and ViPlanner rely on single-frame visual input, which restricts temporal consistency in trajectory planning. This limitation often leads to task failure or inefficient behaviors, particularly in the Turtlebot experiment, where the robot camera has already passed obstacles but the body is not. Then, a severe change in path planning can lead to collision.
- **Robustness to Depth Noise:** Traditional planning-based approaches are highly sensitive to noisy depth sensing. In such cases, global mapping errors can result in overly conservative plans, while short-horizon local maps can also cause planning inconsistencies and collisions.
- **Resistance to Depth Illusions:** As depth-centric methods, both iPlanner and ViPlanner are susceptible to

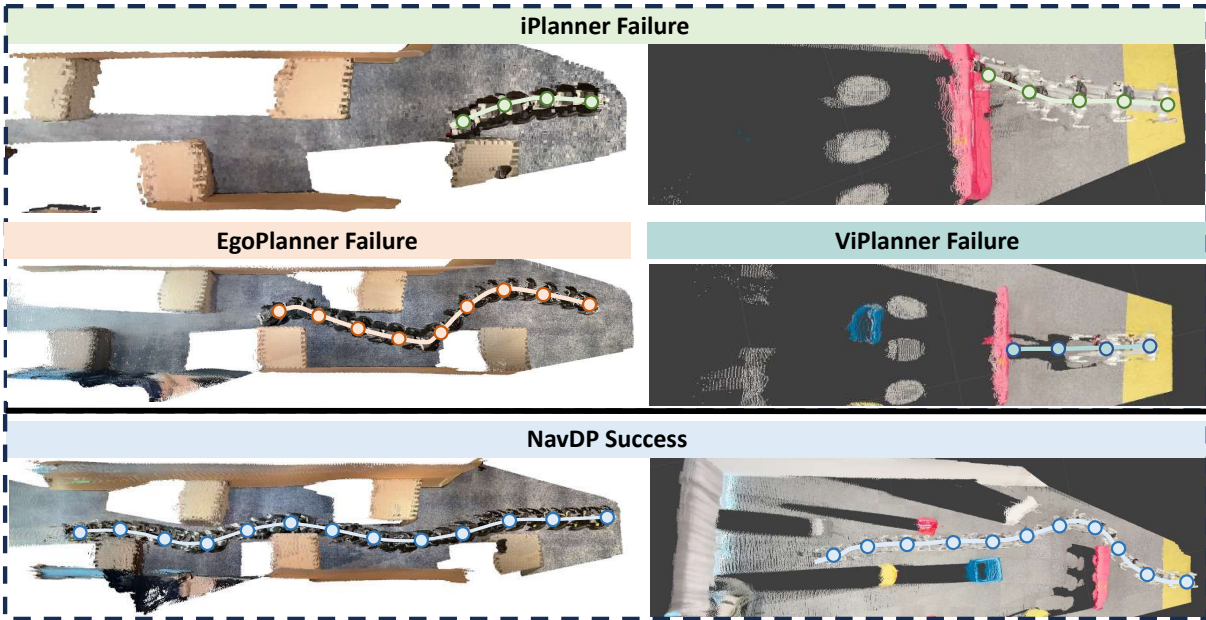


Fig. 4: Visualization of comparison among navigation approaches. Two common failure modes of baselines are displayed: (1) Collision caused by short-horizon memory and planning inconsistency. (2) Failing due to irregular object geometry. Our proposed NavDP can robustly handle both scenarios. The visualization is achieved by *Pi*³ [41].

misleading object geometries. In the Unitree-Go2 experiments, we place obstacles with holes in front of the robot. Both methods failed to interpret the geometry correctly and trying to walk through the obstacles.

Quantitative visualization of the results from different approaches for point-goal navigation task is presented in Figure 4. The detailed navigation process is provided in the supplementary video. Besides, our proposed method also surpasses the baseline methods in no-goal navigation task by a large margin: In simulation, NavDP achieves **2.9x** performance in average exploration time and **3.1x** performance in average exploration area than NoMad in simulation and **3.8x** exploration time in real-world, demonstrating a strong zero-shot generalization to out-of-distribution scenarios.

For **Q3**, we deploy NavDP on the Unitree Go2, Unitree G1 and Galaxea R1 robot and visualize the top-2 generated trajectories with the best critic values as shown in Figure 5. Although the observation views, the existence of pedestrian interference, camera field of view, varying light conditions, the existence of motion blur dramatically make the observation images different from the training dataset, our proposed method still generalize well and can achieve long-horizon navigation without any collision and human intervention over 100 meters. More examples are shown in the appendix video.

For **Q4**, we conduct a comprehensive ablation study on the point-goal navigation benchmark to investigate three critical factors that may influence the overall performance of NavDP: (1) input modalities, (2) the function of the critic prediction, and (3) the choice of training objectives. Specifically, we evaluate six variants of the original NavDP model:

- **w/o Depth***: An RGB-only version of the NavDP model, with the depth input branch removed.
- **w/o RGB***: A depth-only version of the NavDP model,

PointNav	Sim-Home		Sim-Commercial	
	Success	SPL	Success	SPL
w/o Depth*	47.8	44.3	66.1	63.7
w/o RGB*	53.9	49.6	70.3	66.7
w/o Multiframe RGB*	56.9	51.7	72.0	68.2
w/o Selection	53.1	49.0	65.6	62.5
w/o Augmentation*	57.3	52.2	73.4	69.2
w/o No-Goal*	56.8	51.6	73.5	69.9
Original NavDP	60.3	54.7	74.1	70.5

TABLE V: Quantitative results of ablation experiments.

with the RGB input branch removed.

- **w/o Multiframe RGB***: A single-frame RGB-D variant of NavDP, replacing multi-frame RGB input with a single-frame representation.
- **w/o Selection**: A variant of NavDP where the trajectory selection is randomized instead of using the critic-based selection scheme.
- **w/o Augmentation***: A version of NavDP where collision trajectories are excluded during critic head training, disabling the trajectory augmentation scheme.
- **w/o No-Goal***: A variant of NavDP trained without the no-goal-based auxiliary trajectory prediction objectives.

Variants marked with an asterisk (*) indicate that the model is retrained using the same configuration and dataset as the original NavDP. In contrast, variants without an asterisk share the same model weights as the original NavDP. The ablation results are presented in Table V. These results highlight three key technical conclusions:

- RGB-D fusion is essential for more robust navigation

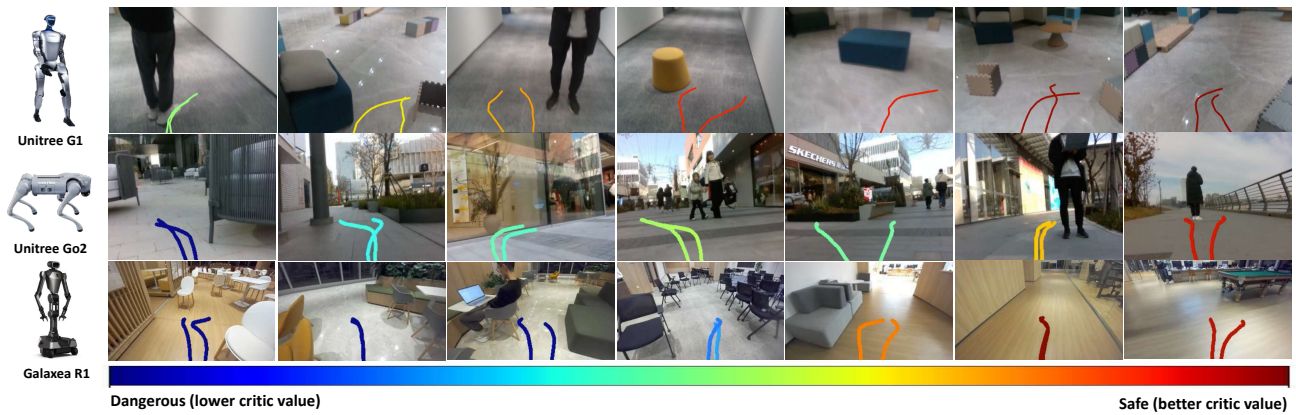


Fig. 5: Trajectory visualization of on different robots. We project the predicted trajectories back to the image space and colorize them according to the corresponding critic values. The **bluer** trajectories indicate higher risk, whereas the **redder** trajectories represent safer paths.

performance. Without the depth as input, the overall performance drops **10.3%** in success rate and without the rgb images as input, the overall performance drops **5.1%**. Multi-frame of RGB input also contributes **2.8%** improvements in success rate.

- Critic function plays an important role for improving the planning safety. With the same model weights but removing the critic-based trajectory selection, the success rate decreases **7.8%** in simulation. Learning critic function from contrastive samples is also important. After removing the trajectory augmentation for critic training, the success drops **3.0%** in home scenes.
- The no-goal task is a useful auxiliary task that can be jointly trained. With the no-goal task training objectives, the overall point-goal navigation performance increase **2.1%** in success rate and **1.8%** in SPL.

For Q5, we train a variant of NavDP model only with the data collected at low robot heights ($< 0.5m$) and camera views to determine whether the domain randomization contributes to the cross-embodiment generalization. Then, we set up two challenging scenarios with cluttered layouts and evaluate the point-goal navigation task performance of two models with both Unitree-Go2 and Galaxea-R1 robots. Each scene is evaluate for 20 episodes for each embodiment. As two robots differ in heights, their path planning strategy should be different: For the Unitree-Go2, it would be more efficient to walk under the table in SceneB, but the Galaxea R1 must take a detour to avoid the table. The quantitative results are shown in Figure 6. We find that without the cross-embodiment data, the detour skill for the table is not exhibited, thus leading to a drop in success rate from **90%** to **20%**, while the overall performance on Unitree-Go2 is maintained.

VI. CONCLUSION & FUTURE WORKS

In this paper, we introduce a novel navigation diffusion policy (NavDP) that achieves strong zero-shot sim-to-real and cross-embodiment generalization performance. Our policy demonstrates real-time path-planning and collision-avoidance abilities under both static and dynamic scenarios.

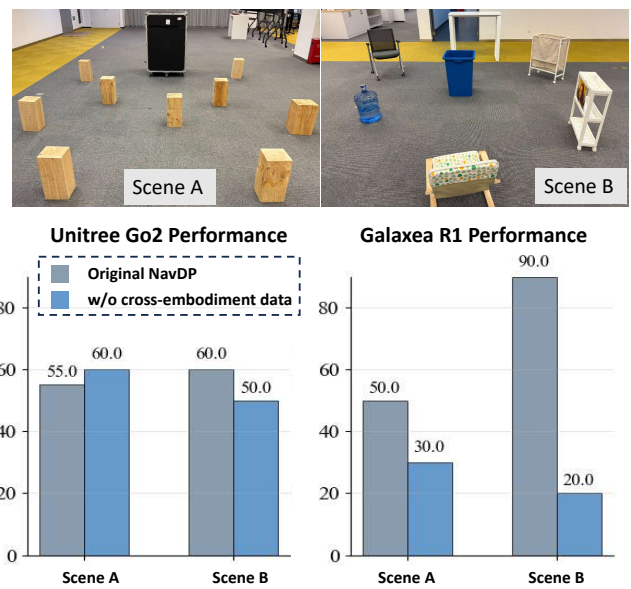


Fig. 6: Ablation study result on cross-embodiment dataset. The top row demonstrates the evaluation scenes while the bottom shows the performance. Without the cross-embodiment data, the performance on Galaxea R1 drops **70%** in success rate.

Two key ingredients contributes to the NavDP performance. The first is our proposed automatic data generation pipeline and large-scale simulation navigation dataset. The second is the design of the entire network with efficient RGB-D fusion and contrastive critic training. Our NavDP provides a novel perspective in building versatile end-to-end navigation policies and a strong backbone for further improvements.

For future work, we plan to explore efficient post-training strategies to further enhance performance—an essential step for real-world deployment. Additionally, we aim to extend NavDP to support a wider range of navigation goals, particularly those expressed through natural language instructions. Finally, integrating a global memory mechanism to enable long-term exploration and more holistic navigation behavior represents another promising direction for future research.

REFERENCES

- [1] N. Hirose, F. Xia, R. Martín-Martín, A. Sadeghian, and S. Savarese, “Deep visual mpc-policy learning for navigation,” *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3184–3191, 2019.
- [2] H. Karnan, A. Nair, X. Xiao, G. Warnell, S. Pirk, A. Toshev, J. Hart, J. Biswas, and P. Stone, “Socially compliant navigation dataset (scand): A large-scale dataset of demonstrations for social navigation,” *IEEE Robotics and Automation Letters*, 2022.
- [3] N. Hirose, D. Shah, A. Sridhar, and S. Levine, “Sacson: Scalable autonomous control for social navigation,” *IEEE Robotics and Automation Letters*, 2023.
- [4] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma *et al.*, “The replica dataset: A digital replica of indoor spaces,” *arXiv preprint arXiv:1906.05797*, 2019.
- [5] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang, “Matterport3d: Learning from rgb-d data in indoor environments,” *arXiv preprint arXiv:1709.06158*, 2017.
- [6] H. Fu, B. Cai, L. Gao, L.-X. Zhang, J. Wang, C. Li, Q. Zeng, C. Sun, R. Jia, B. Zhao *et al.*, “3d-front: 3d furnished rooms with layouts and semantics,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10933–10942.
- [7] M. Khanna, Y. Mao, H. Jiang, S. Haresh, B. Shacklett, D. Batra, A. Clegg, E. Undersander, A. X. Chang, and M. Savva, “Habitat synthetic scenes dataset (hssd-200): An analysis of 3d scene scale and realism tradeoffs for objectgoal navigation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 16384–16393.
- [8] S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang *et al.*, “Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai,” *arXiv preprint arXiv:2109.08238*, 2021.
- [9] H. Wang, J. Chen, W. Huang, Q. Ben, T. Wang, B. Mi, T. Huang, S. Zhao, Y. Chen, S. Yang *et al.*, “Grutopia: Dream general robots in a city at scale,” *arXiv preprint arXiv:2407.10943*, 2024.
- [10] X. Meng, X. Yang, S. Jung, F. Ramos, S. S. Jujjavarapu, S. Paul, and D. Fox, “Aim my robot: Precision local navigation to any object,” *IEEE Robotics and Automation Letters*, 2025.
- [11] A. Sridhar, D. Shah, C. Glossop, and S. Levine, “Nomad: Goal masked diffusion policies for navigation and exploration,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 63–70.
- [12] K.-H. Zeng, Z. Zhang, K. Ehsani, R. Hendrix, J. Salvador, A. Herrasti, R. Girshick, A. Kembhavi, and L. Weihs, “Poliformer: Scaling on-policy rl with transformers results in masterful navigators,” in *8th Annual Conference on Robot Learning*.
- [13] A. Eftekhari, L. Weihs, R. Hendrix, E. Caglar, J. Salvador, A. Herrasti, W. Han, E. VanderBil, A. Kembhavi, A. Farhadi *et al.*, “The one ring: a robotic indoor navigation generalist,” *arXiv preprint arXiv:2412.14401*, 2024.
- [14] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” in *Proceedings of Robotics: Science and Systems (RSS)*, 2023.
- [15] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki, “3d diffuser actor: Policy diffusion with 3d scene representations,” *arXiv preprint arXiv:2402.10885*, 2024.
- [16] X. Li, V. Belagali, J. Shang, and M. S. Ryoo, “Crossway diffusion: Improving diffusion-based visuomotor policy via self-supervised learning,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 16841–16849.
- [17] Y. Wang, Y. Zhang, M. Huo, R. Tian, X. Zhang, Y. Xie, C. Xu, P. Ji, W. Zhan, M. Ding *et al.*, “Sparse diffusion policy: A sparse, reusable, and flexible policy for robot learning,” *arXiv preprint arXiv:2407.01531*, 2024.
- [18] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, “3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations,” in *Proceedings of Robotics: Science and Systems (RSS)*, 2024.
- [19] A. Prasad, K. Lin, J. Wu, L. Zhou, and J. Bohg, “Consistency policy: Accelerated visuomotor policies via consistency distillation,” *arXiv preprint arXiv:2405.07503*, 2024.
- [20] Z. Wang, Z. Li, A. Mandelkar, Z. Xu, J. Fan, Y. Narang, L. Fan, Y. Zhu, Y. Balaji, M. Zhou *et al.*, “One-step diffusion policy: Fast visuomotor policies via diffusion distillation,” *arXiv preprint arXiv:2410.21257*, 2024.
- [21] X. Huang, Y. Chi, R. Wang, Z. Li, X. B. Peng, S. Shao, B. Nikolic, and K. Sreenath, “Diffuselo: Real-time legged locomotion control with diffusion from offline datasets,” in *8th Annual Conference on Robot Learning*, 2024.
- [22] J. Zhang, M. Wu, and H. Dong, “Generative category-level object pose estimation via diffusion models,” in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
- [23] M. Xu, Z. Xu, C. Chi, M. Veloso, and S. Song, “Xskill: Cross embodiment skill discovery,” in *Conference on Robot Learning*. PMLR, 2023, pp. 3536–3555.
- [24] C. Wang, H. Shi, W. Wang, R. Zhang, L. Fei-Fei, and C. K. Liu, “Dexcap: Scalable and portable mocap data collection system for dexterous manipulation,” *arXiv preprint arXiv:2403.07788*, 2024.
- [25] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine, “Gnm: A general navigation model to drive any robot,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7226–7233.
- [26] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine, “Vint: A foundation model for visual navigation,” *arXiv preprint arXiv:2306.14846*, 2023.
- [27] F. Yang, C. Wang, C. Cadena, and M. Hutter, “iplanner: Imperative path planning,” *arXiv preprint arXiv:2302.11434*, 2023.
- [28] P. Roth, J. Nubert, F. Yang, M. Mittal, and M. Hutter, “Viplanner: Visual semantic imperative learning for local navigation,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 5243–5249.
- [29] W. Cai, S. Huang, G. Cheng, Y. Long, P. Gao, C. Sun, and H. Dong, “Bridging zero-shot object navigation and foundation models through pixel-guided navigation skill,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 5228–5234.
- [30] K. Ehsani, T. Gupta, R. Hendrix, J. Salvador, L. Weihs, K.-H. Zeng, K. P. Singh, Y. Kim, W. Han, A. Herrasti *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*, 2024, pp. 16238–16250.
- [31] J. Zhang, K. Wang, R. Xu, G. Zhou, Y. Hong, X. Fang, Q. Wu, Z. Zhang, and H. Wang, “Navid: Video-based vlm plans the next step for vision-and-language navigation,” *Robotics: Science and Systems*, 2024.
- [32] J. Zhang, K. Wang, S. Wang, M. Li, H. Liu, S. Wei, Z. Wang, Z. Zhang, and H. Wang, “Uni-navid: A video-based vision-language-action model for unifying embodied navigation tasks,” *arXiv preprint arXiv:2412.06224*, 2024.
- [33] A.-C. Cheng, Y. Ji, Z. Yang, X. Zou, J. Kautz, E. Bıyık, H. Yin, S. Liu, and X. Wang, “Navila: Legged robot vision-language-action model for navigation,” *arXiv preprint arXiv:2412.04453*, 2024.
- [34] M. Denninger, M. Sundermeyer, D. Winkelbauer, D. Olefir, T. Hodan, Y. Zidan, M. Elbadrawy, M. Knauer, H. Katam, and A. Lodhi, “Blenderproc: Reducing the reality gap with photorealistic rendering,” in *16th Robotics: Science and Systems, RSS 2020, Workshops*, 2020.
- [35] F. Xia, A. R. Zamir, Z.-Y. He, A. Sax, J. Malik, and S. Savarese, “Gibson Env: real-world perception for embodied agents,” in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.
- [36] D. Shah, B. Eysenbach, G. Kahn, N. Rhinehart, and S. Levine, “Rapid exploration for open-world navigation with latent goal models,” *arXiv preprint arXiv:2104.05859*, 2021.
- [37] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, “Depth anything v2,” *Advances in Neural Information Processing Systems*, vol. 37, pp. 21875–21911, 2024.
- [38] J. Ho, A. Jain, and P. Abbeel, “Denosing diffusion probabilistic models,” *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.
- [39] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra, “Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames,” in *International Conference on Learning Representations*, 2019.
- [40] X. Zhou, Z. Wang, H. Ye, C. Xu, and F. Gao, “Ego-planner: An esdf-free gradient-based local planner for quadrotors,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 478–485, 2020.
- [41] Y. Wang, J. Zhou, H. Zhu, W. Chang, Y. Zhou, Z. Li, J. Chen, J. Pang, C. Shen, and T. He, “ π^3 : Scalable permutation-equivariant visual geometry learning,” 2025. [Online]. Available: <https://arxiv.org/abs/2507.13347>