

RaFM-Nav: A Risk-Aware Flow Matching Framework for Collision-Free Visual Navigation

Liang Li¹, Wenbo Wang¹, Jianjian Jiang¹, Jinke Meng^{1,*}, Wei-Shi Zheng^{1,2}

Abstract—Visual navigation policies trained on human demonstrations can effectively leverage visual inputs to reach target destinations, however, they may exhibit unsafe behaviors in cluttered environments, as they lack explicit safety-related modeling and thus fail to capture safety knowledge from human demonstrations. To mitigate this, we present RaFM-Nav, a risk-aware visual navigation framework that enhances safety in complex scenes via two complementary mechanisms. First, we propose an Obstacle Perception Grid (OPG) module that transforms monocular depth into a top-down occupancy grid to provide obstacle occupancy information, thereby explicitly modeling potential scene risks at the perception level. Following that, we introduce a group-relative trajectory alignment strategy in the fine-tuning phase, which evaluates in-group trajectories via proposed safety metrics to guide the model toward safer trajectory modes at the supervision level without increasing inference-time computation. We validate RaFM-Nav on a real-world quadruped robot. In densely cluttered environments, our method demonstrates exceptional robustness compared to the previous state-of-the-art methods, more than doubling the navigation success rate (30.0% to 70.0%) while slashing the collision frequency by over $2\times$ in both goal-conditioned navigation and goal-free exploration tasks.

I. INTRODUCTION

Visual navigation is important for autonomous mobile robots operating in real-world environments [1]–[3]. Given visual observations, an embodied agent perceives its surroundings and generates actions to reach a goal safely.

Previous learning-based navigation methods directly map visual observations to actions [4]–[10]. While effective, these early approaches mainly produce deterministic actions, which limits their ability to capture multimodal action distributions as illustrated in [11]. Recent works leverage generative paradigms, such as diffusion or flow matching, to model multiple plausible trajectories [8], [11]–[14].

While these methods can effectively fit human demonstration data to improve success rates, they may still exhibit unsafe behaviors. This stems from the fact that although human demonstrations implicitly contain safety-related cues, the demonstrations are often confounded by diverse individual preferences and varying environmental contexts. Consequently, imitation learning-based policies fail to extract a robust safety distribution within the latent space, resulting in action outputs that remain prone to collision risks, as

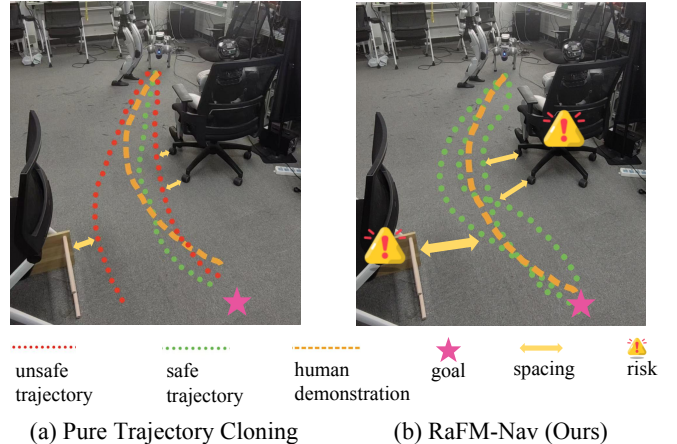


Fig. 1: Illustration comparing the trajectory distributions generated by different policies. When deployed in unseen environments, standard generative imitation policies tend to yield highly variable trajectory distributions, leaving the agent vulnerable to a wide range of unsafe paths with marginal obstacle clearances (left). By explicitly modeling spatial risks, our method effectively concentrates and shifts the generated distribution toward safer, well-centered traversals with increased safety margins (right).

illustrated in Fig. 1. To address this issue, we propose **RaFM-Nav**, a **R**isk-aware **F**low **M**atching framework for safe navigation in densely cluttered environments. The key insight of our framework is to leverage two complementary mechanisms to explicitly model navigation safety: a perception-level grid that models obstacle occupancy to enhance risk awareness, and a supervision-level loss inspired by Group-Relative Policy Optimization (GRPO) [15] that guides the policy toward safer trajectory modes.

Specifically, we propose an Obstacle Perception Grid (OPG) module, which first back-projects the estimated depth map into a 3D point cloud and subsequently discretizes the local region into a structured 2D spatial grid. Each cell in this grid contains point density and normalized height, offering spatial cues regarding obstacle occupancy and free-space layout. By integrating these grid-based embeddings with visual tokens, our OPG module provides additional spatial priors for risk regions and enhances safety awareness at the perception level.

Complementary to this perception-level enhancement, we further propose group-relative trajectory alignment, a fine-tuning strategy inspired by GRPO [15]. Our framework sam-

¹Sun Yat-sen University.

²Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China.

*Corresponding author.

This work was supported partially by NSFC (92470202), Guangdong NSF Project (No. 2024A1515010101, No. 2023B1515040025), Guangdong Key Research and Development Program (No. 2024B0101040004, No. 2025B0909020002).

ples a group of candidate trajectories and ranks them using a safety-oriented reward function. This reward integrates a collision penalty that penalizes trajectories passing too close to detected obstacles, with an open-space incentive that encourages the agent to prioritize wider roads when multiple feasible paths exist, effectively biasing the policy to align the action distribution with safer modes. It should be noted that the alignment only occurs at the supervision level during training, ensuring that the agent maintains high inference efficiency while reducing collision risks in complex scenes.

Extensive real-world experiments conducted with a Uni-tree Go2 quadruped robot across three environments comprising static indoor clutter, unstructured outdoor clutter and dynamic scenarios with sudden pedestrian interference demonstrate the following: (1) **Effectiveness**: In goal-conditioned navigation, RaFM-Nav achieves a 70% success rate, significantly outperforming the SOTA method. (2) **Safety**: In goal-conditioned navigation, our method reduces the average collisions per run to 1.1, compared to 2.5 for the SOTA method. Furthermore, in goal-free exploration tasks, our approach maintains a superior safety margin with only 1.0 collision per run, representing a 70% reduction relative to SOTA methods.

In summary, the main contributions of this work are:

- **Obstacle Perception Grid**: We propose the Obstacle Perception Grid, a structured spatial representation obtained via 3D back-projection, which provides explicit spatial grounding and enables the agent to identify risk-prone regions at the perception stage.
- **Group-Relative Trajectory Alignment**: We introduce Group-Relative Trajectory Alignment, a supervision-level fine-tuning strategy inspired by GRPO that biases navigation policies toward safer action modes with our proposed safety-oriented reward function.
- **Real-World Validation**: We provide extensive real-world experiments on a quadruped robot across diverse environments, demonstrating that RaFM-Nav significantly boosts success rates to 70% while reducing collision frequency by up to 70% compared to state-of-the-art methods.

II. RELATED WORK

A. Visual Navigation

Classical autonomous navigation typically relies on SLAM and geometric planning pipelines, which have demonstrated strong robustness in structured environments and large-scale deployments [5], [16], [17].

With the rise of large-scale robot datasets and deep representation learning, end-to-end visual navigation has emerged as an alternative paradigm that maps egocentric observations to control actions [18]. Goal-conditioned policies trained across diverse embodiments improve generalization by learning normalized action representations [19], [20].

To capture the multimodal nature of navigation behavior, recent methods model distributions over future trajectories rather than predicting a single deterministic action [21].

Diffusion- and flow-matching-based policies have shown strong performance in this setting, supporting both goal-conditioned navigation and open-ended exploration [8], [14].

ViNT generates subgoal images for long-horizon navigation [20], while NoMaD applies diffusion directly in the action space for both goal-directed and exploratory behaviors [8]. Flow matching offers a more efficient alternative to diffusion by learning continuous transport dynamics for action generation [22], [23]. While these generative methods can effectively fit human demonstration data to improve success rates, they may still exhibit unsafe behaviors.

Recent works have begun to explore safety in generative visual navigation. SaferPath [24] improves safety by selecting the safety trajectories generated by diffusion policies through a post-hoc manner and does not improve the policy to internalize safety-aware behaviors. Self-Imitated Diffusion Policy (SIDP) [25] improves diffusion-based navigation via a self-imitation training scheme that iteratively selects safety-aware trajectories and distills them back into the policy. Its reward is computed with an indicator-style formulation, yielding a relatively hard signal that often leads to sparse or unstable gradients during policy fine-tuning.

The proposed RaFM-Nav differs from the prior safety-oriented policy SIDP in three aspects. First, we introduce a perception-level occupancy grid that explicitly models obstacle geometry to enhance risk awareness. Second, we design a soft, safety-oriented reward to guide the policy toward safer trajectory modes. Third, we incorporate an open-space incentive that encourages the agent to prefer wider corridors when multiple feasible paths exist.

B. Offline Preference Alignment

Alignment approaches based on online reinforcement learning [26]–[28] have proven highly effective for policy optimization, though they typically require repeated environment interactions to collect reward or preference signals.

For real-world robotic systems, online interactions are often unavailable, as deploying them in physical environments for extensive data collection is impractical. Therefore, offline preference alignment strategy [29], [30] is introduced to fine-tune a policy using offline data, adapting its behavior to match different preference objectives.

In robotics, offline preference alignment has emerged as a versatile paradigm for incorporating task-specific objectives beyond standard imitation learning.

Prior work has leveraged ranking-based objectives to guide trajectory selection toward desirable outcomes, such as higher task completion rates or shorter execution times [21], [31]. Within diffusion-based policy learning, preference formulations have been employed to shape generated action sequences toward smoother, more energy-efficient, or human-interpretable behaviors, thereby improving generation fidelity beyond behavioral cloning [11], [32]. More recently, preference learning has been extended to enforce high-level behavioral constraints, including navigation styles, spatial compliance, and language-conditioned instructions [33], [34].

Unlike traditional policy gradient methods that rely on a critic network, GRPO [15] estimates advantages through comparisons within a group of sampled responses, which is particularly effective when multiple candidates are available. This group-wise comparison mechanism enables more stable policy optimization while mitigating the risk of overfitting to noisy reward signals [35], [36]. Inspired by group-relative policy optimization, we introduce a safety-aware alignment strategy for generative visual navigation. We design a safety-oriented reward that penalizes obstacle proximity and rewards open-space traversal. By optimizing the policy against the relative rankings of sampled trajectories based on these safety-oriented rewards, our policy explicitly steers the generative distribution away from high-risk paths, enabling robust collision avoidance.

III. METHODOLOGY

A. Problem Formulation and Overview

Problem Formulation. We study the goal-conditioned visual navigation problem. At time t , the robot observes an RGB image o_t and is given a goal image o_g . To capture short-term temporal context, we use an observation history $O_t = \{o_{t-T_o+1}, \dots, o_t\}$ of length T_o . The policy outputs an ego-centric waypoint trajectory $A_t = \{a_t, \dots, a_{t+T_a-1}\}$ with horizon T_a , where each waypoint is represented by 2D spatial coordinates in the robot’s local ground plane. Our objective is to learn a risk-aware conditional generative policy $\pi_\theta(A_t | O_t, o_g)$ that preserves goal-reaching capability while generating safe waypoints in densely cluttered environments. **Overview.** As shown in Fig. 2, RaFM-Nav is a goal conditioned visual navigation framework that learns a risk aware policy over future waypoint trajectories from a short history of RGB observations and a goal image, using conditional flow matching as the trajectory generation backbone. In the initial training phase, in addition to RGB and depth features, we introduce an Obstacle Perception Grid (OPG) that represents the free space layout in front of the agent. These three modalities are jointly utilized as conditional inputs to train a base flow matching policy via behavioral cloning. During fine-tuning, we aim to improve risk awareness of the policy by introducing a safety-oriented reward that explicitly assesses trajectory safety. We achieve this using group-relative trajectory alignment. Concretely, we sample a group of candidate trajectories, rank them by the safety reward, and perform a group-relative update that assigns higher weight to trajectories that are safer than others within the same group.

B. Conditional Flow Matching with Obstacle Perception Grid

Following FlowNav [14], we adopt conditional flow matching (CFM) as our basic generative backbone for trajectory prediction. The policy predicts future waypoints conditioned on visual observations, using an observation encoder that combines RGB images with implicit depth features. To further improve obstacle awareness, we additionally learn an Obstacle Perception Grid (OPG) from the estimated depth map to represent the free-space layout in front of

the agent. Then we utilize the RGB observations, implicit depth features, and the Obstacle Perception Grid jointly as conditioning inputs, and train the flow matching policy with behavior cloning.

Obstacle Perception Grid (OPG). We learn a structured front-view free-space map, termed Obstacle Perception Grid (OPG), that explicitly encodes the surrounding free-space layout in front of the agent. This explicit spatial representation provides direct obstacle cues for policy learning, complementing the implicit obstacle features from RGB and depth information.

Specifically, given the relative depth map extracted from RGB images, we back-project it into a 3D point cloud using camera intrinsics $K \in R^{3 \times 3}$. We discretize the local ground region in front of the robot into a fixed-resolution horizontal grid of size $G \times G$, which is set to 16×16 in our implementation. The grid captures the local space with a lateral axis x and a forward depth axis z . For each grid cell i , we compute four geometric statistics that jointly characterize the free-space layout and potential obstacles:

- Point density ρ_i : the number of 3D points falling into cell i , which is further processed as $\log(1 + \rho_i)$ to dampen outliers and indicate spatial occupancy;
- Depth range $(z_{\min,i}, z_{\max,i})$: the normalized minimum and maximum depth values in cell i , representing the closest observed surface and the farthest visible boundary, respectively;
- Normalized maximum height $y_{\max,i}$: the maximum height value in cell i , normalized to $[0, 1]$. This is particularly important for obstacle detection: higher values indicate the presence of obstacles (e.g., furniture, pedestrians) standing above the ground plane.

Then, we embed the obstacle representation of cell i into a spatial token:

$$\mathbf{x}_{\text{opg}}^i = \text{MLP}([\log(1 + \rho_i), z_{\min,i}, z_{\max,i}, y_{\max,i}]), \quad (1)$$

where MLP denotes a two-layer perceptron. Collectively, all grid tokens form the obstacle representation $\mathbf{X}_{\text{opg}} \in R^{(G \times G) \times D}$, which provides a camera-centric top-down prior that makes free-space layout explicitly accessible.

In addition to the OPG, following prior work [14], we encode the observation history into temporal visual tokens. Specifically, historical ego-centric RGB frames are processed by an EfficientNet-B0 encoder, and the resulting embeddings are projected into D -dimensional tokens, yielding $\mathbf{X}_{\text{obs}} \in R^{T_o \times D}$. Meanwhile, a depth-aware token $\mathbf{X}_{\text{depth}} \in R^{1 \times D}$ is extracted from the current RGB observation using the frozen DINOv2 backbone of Depth Anything V2.

CFM Policy Learning. We utilize the RGB observations, implicit depth features, and the Obstacle Perception Grid jointly as conditioning inputs, and train the flow matching policy with behavior cloning.

Then, all perception features are concatenated and processed by a Transformer encoder to produce the final conditioning context $\mathbf{h}_t$:

$$\mathbf{h}_t = \text{Transformer}([\mathbf{X}_{\text{obs}}, \mathbf{X}_{\text{depth}}, \mathbf{X}_{\text{opg}}, \mathbf{X}_{\text{goal}}]), \quad (2)$$

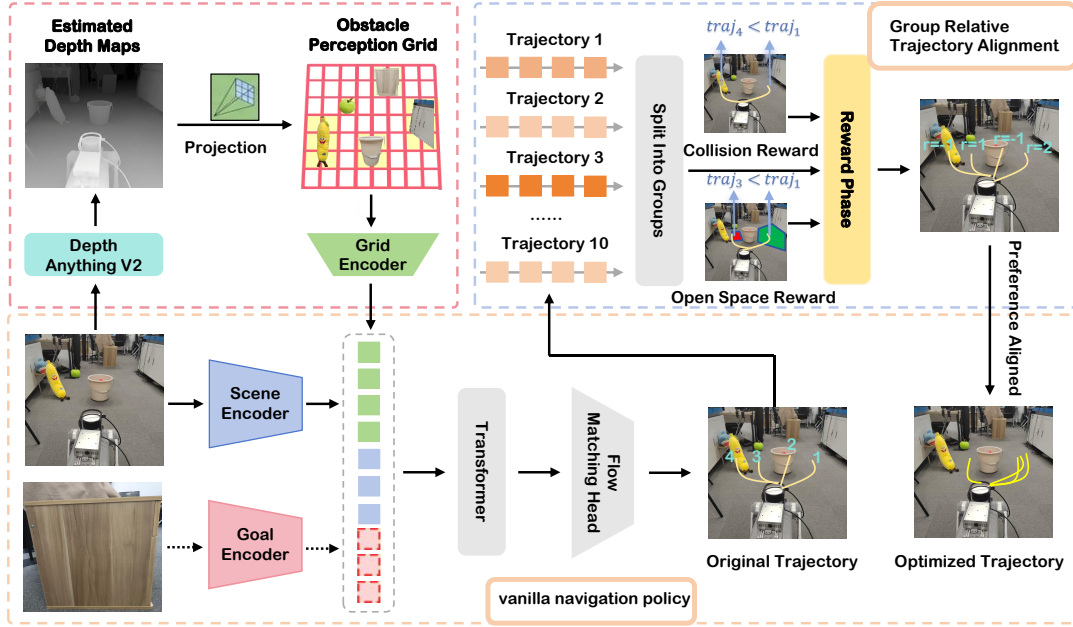


Fig. 2: **Overview of the RaFM-Nav architecture.** Building upon a standard flow-matching navigation backbone, we introduce Obstacle Perception Grid (OPG) to extract explicit spatial layout features from estimated depth maps, augmenting the standard scene and optional goal tokens. During the fine-tuning phase, the group relative trajectory alignment mechanism evaluates the sampled trajectories using safety-oriented rewards (collision penalty and open-space reward) to shift the generative policy toward safer, collision-free behaviors.

where $\mathbf{X}_{\text{goal}} \in R^{1 \times D}$ is the goal embedding. Given the conditioning context $\mathbf{h}_t$, we adopt conditional flow matching (CFM) to generate waypoint sequences. The context $\mathbf{h}_t$ serves as the conditioning input to the flow model, which learns to transport noise samples to the target waypoints. The flow model $v_\theta(\tau, \mathbf{A}_\tau, \mathbf{h}_t)$, which parameterizes our generative policy $\pi_\theta(A_t | O_t, g)$ in our problem formulation, is trained by minimizing:

$$\mathcal{L}_{\text{CFM}} = E_{\tau, \mathbf{A}_0, \mathbf{A}_1} \left[\left\| v_\theta(\tau, \mathbf{A}_\tau, \mathbf{h}_t) - (\mathbf{A}_1 - \mathbf{A}_0) \right\|^2 \right], \quad (3)$$

where $\tau \in [0, 1]$ is the time variable along the flow path, $\mathbf{A}_1 \in R^{T_a \times 2}$ denotes the future waypoint sequence with horizon T_a , and $\mathbf{A}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the noise. The interpolation path is $\mathbf{A}_\tau = (1 - \tau)\mathbf{A}_0 + \tau\mathbf{A}_1$.

C. RL Fine-tuning via Group-Relative Trajectory Alignment

Imitation learning based policies mainly learn to reproduce expert behavior from the training data, without explicitly encoding the underlying safety constraints. Consequently, a single observation may admit multiple behaviorally plausible trajectories, but these candidates can differ in their clearance to nearby obstacles. Without an explicit preference for safety constraints, the policy may select trajectories with unnecessarily small clearance, leading to suboptimal obstacle avoidance when deployed in novel environments. We therefore fine-tune the cloned policy with group-relative trajectory alignment by sampling a group of candidate trajectories, ranking them with a safety oriented reward, and applying a group-relative update to increase the likelihood of higher ranked safety behaviors.

Safety-Oriented Reward. We first propose a safety-oriented reward to evaluate the safety of each trajectory in a sampled set. It combines two complementary signals: the obstacle collision penalty $r_{\text{collision}}$ penalizes trajectories that incur higher collision risk by passing too close to obstacles, and the open-space reward r_{open} encourages the policy to prefer traversing locally open regions when multiple feasible passages exist. The reward is formulated as:

$$R(A) = r_{\text{open}}(A) - \lambda \cdot r_{\text{collision}}(A), \quad (4)$$

where A denotes a trajectory and $\lambda > 0$ is a coefficient balancing the collision cost.

The obstacle collision penalty is defined as

$$r_{\text{collision}}(A) = \max_{m \in \{1, \dots, T_a\}} p_m(A), \quad (5)$$

where T_a is the prediction horizon and $p_m(A)$ denotes the penalty at the m -th waypoint. For each waypoint m along the trajectory A , the penalty is defined as

$$p_m(A) = \begin{cases} \left(1 - \frac{d_m(A)}{d_{\text{safe}}}\right)^2, & \text{if } d_m(A) < d_{\text{safe}}, \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

where $d_m(A)$ denotes the minimum distance from the m -th waypoint to nearby obstacles and d_{safe} is the safety distance threshold.

The open-space reward r_{open} encourages the policy to prefer traversing locally open regions rather than narrow passages, and is formulated as:

$$r_{\text{open}}(A) = \mathbf{v}_{\text{open}}^\top \mathbf{v}_{\text{traj}} \cdot \exp(-\gamma \cdot r_{\text{collision}}(A)), \quad (7)$$

where $\mathbf{v}_{\text{open}}$ denotes the unit direction vector toward the most open traversable direction in front of the agent as indicated by the obstacle perception grid in Eq. (1), and it is used as a supervision signal to bias trajectory generation toward locally more open space. The $\mathbf{v}_{\text{traj}}$ is the normalized direction vector of the generated trajectory A , and the dot product computes the cosine similarity, directly maximizing the reward when the trajectory perfectly aligns with the locally most open space. To prevent the agent from receiving reward simply by pointing toward open space while still being too close to obstacles, we apply $r_{\text{collision}}$ as a continuous safety gate that reduces the reward when collision risk is high, a larger γ gives more penalty when the robot is near obstacles, even if it is moving toward open space.

Policy Fine-Tuning via Group-Relative Trajectory Alignment. To refine the cloned policy toward safer behaviors, we adopt a group-relative alignment inspired by GRPO [15]. For each observation, we sample a group of K candidate trajectories and evaluate them using the proposed safety reward $R(A)$ defined above.

For each group of K trajectories sampled from the same observation, we compute the group-relative advantage:

$$\tilde{R}_j = \frac{R(A_j) - \mu}{\sigma}, \quad (8)$$

where μ and σ are the mean and standard deviation of rewards within the group. These advantages are then transformed into softmax weights:

$$w_j = \frac{\exp(\beta \tilde{R}_j)}{\sum_{l=1}^K \exp(\beta \tilde{R}_l)}, \quad (9)$$

where β controls the sharpness of preference weighting.

These weights bias the conditional flow matching objective:

$$\mathcal{L}_{\text{align}} = E_{j,\tau} \left[w_j \|v_{\theta}(\tau, \mathbf{A}_{\tau,j}, \mathbf{h}_t) - (\mathbf{A}_{1,j} - \mathbf{A}_{0,j})\|^2 \right], \quad (10)$$

where $\mathbf{A}_{\tau,j}$ denotes the τ -time step along the flow path between noise $\mathbf{A}_{0,j}$ and the trajectory tensor $\mathbf{A}_{1,j}$ corresponding to the sampled trajectory A_j . To prevent the policy from deviating too far from the expert data, we jointly optimize the alignment loss with the original behavior cloning objective. The final fine-tuning loss is formulated as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CFM}} + \alpha \mathcal{L}_{\text{align}}, \quad (11)$$

where α is a weighting hyperparameter that controls the contribution of the alignment objective.

IV. EXPERIMENTS

A. Implementation Details

For CFM policy learning, we adopt the datasets used in previous work [14], including Go Stanford [37], RECON [38], TartanDrive [39], SACSoN [40], and SCAND [41]. For the RL fine-tuning stage, we utilize MuSoHu [42], SCAND, and GND [43]. Synchronized 2D obstacle annotations are generated from the corresponding



Fig. 3: Evaluation environments. **Left:** indoor scene with dense static obstacles. **Middle:** outdoor environment with natural terrain. **Right:** indoor scenario where an unexpected obstacle is introduced at close range (~ 0.5 m) during robot motion to test emergency avoidance capability.

laser scan measurements and are used exclusively for offline safety reward computation.

All models are initialized from the publicly released FlowNav checkpoint and fine-tuned for 10 epochs using the AdamW optimizer with an initial learning rate of 10^{-4} , a cosine decay schedule with warmup, and a batch size of 128. To preserve pretrained representations and prevent overfitting, we freeze all perception backbones, specifically EfficientNet, DINOv2, and Depth Anything V2. We only update the four-layer Transformer encoder with a hidden dimension of 512, the Obstacle Perception Grid module, and the flow head, which collectively account for approximately 11% of the total parameters. The Obstacle Perception Grid constructs a 16×16 grid covering a local region that spans 4 m laterally and 8 m longitudinally, utilizing dataset-specific camera intrinsics for 3D back-projection.

For offline alignment, we sample $K = 10$ trajectories per context and set the safety gating strength to $\gamma = 5.0$, advantage sharpness to $\beta = 2.0$, weighting hyperparameter to $\alpha = 0.3$, and collision penalty coefficient to $\lambda = 1.0$. Privileged 2D obstacle annotations from synchronized laser scans are strictly used for computing offline safety rewards during training and are completely excluded at inference time. All experiments are conducted on a single NVIDIA RTX 3090 GPU.

B. Experiment Setup

We evaluate RaFM-Nav on real-world goal-conditioned navigation for reaching a visual target as well as goal-free exploration to ensure safe movement without a specified target. We compare our approach against two state-of-the-art generative baselines: NoMaD [8] and FlowNav [14].

The experimental setup comprised three real-world environments with distinct characteristics (Fig. 3): **left**, an indoor environment with static obstacles; **middle**, an outdoor environment featuring natural, unstructured terrain; and **right**, an indoor dynamic environment involving the sudden insertion of a stationary pedestrian approximately 0.5 m in front of the robot’s path to evaluate its emergency avoidance capabilities. Each method is evaluated with 10 independent runs in each environment, 30 runs in total. The reported results are averaged across the three environments.

We report the mean success rate (SR), defined as the percentage of runs that successfully reach the goal without becoming stuck due to collisions. We additionally report

TABLE I: Comparison with SOTA generative methods under the setting of the goal-conditioned navigation and goal-free exploration.

Method	Navigation		Exploration	
	SR (%) $\uparrow$	Coll./run $\downarrow$	Coll./run $\downarrow$	Latency (ms) $\downarrow$
NoMaD [8]	20.0	2.8	3.6	62.1
FlowNav [14]	30.0	2.5	3.5	63.8
Ours-without-alignment	30.0	2.1	2.5	65.0
Ours-without-OPG	40.0	1.6	1.8	63.8
Ours (Full)	70.0	1.1	1.0	65.0

TABLE II: Ablation of reward components in navigation mode.

Method	SR (%) $\uparrow$	Coll./run $\downarrow$
Initial policy (BC)	30.0	2.1
BC+ Open-space reward	40.0	1.8
BC+ Safety penalty	50.0	1.4
Ours (Full)	70.0	1.2

the mean number of collisions per run (Coll./run) as a safety metric and the average inference latency (Latency) per forward pass to evaluate deployment efficiency.

C. Real-World Deployment Setup

All experiments are conducted on a Unitree Go2 quadruped robot, controlled via velocity commands. Inference runs on an external workstation equipped with an RTX 4060 GPU, at 10 Hz with an average forward-pass latency of 65 ms. At each timestep, the robot observes RGB images and predicts a sequence of $T_a = 8$ future ego-centric waypoints, which are converted into velocity commands.

a) Comparison with SOTA Generative Methods: Table I shows that our method consistently outperforms prior SOTA generative baselines under both goal-conditioned navigation and goal-free exploration settings. In navigation, NoMaD and FlowNav reach only 20% and 30% success rates, suffering from high collision rates (2.8 and 2.5 collisions per run, respectively). The suboptimal performance of these baselines stems from their reliance on pure behavior cloning; without explicit geometric grounding or safety constraints, they are prone to generating boundary-skimming trajectories that remain highly vulnerable to compounding errors in dense clutter. In contrast, our full model, RaFM-Nav, actively maintains spatial safety margins by injecting explicit risk awareness into the generative process. Crucially, our method boosts the success rate to 70% primarily by mitigating collision-induced failures.

Similarly, in goal-free exploration, our model achieves the lowest collision rate at 1.0, demonstrating vastly superior generalization and safety compared to NoMaD (3.6) and FlowNav (3.5).

b) Ablation Study: We evaluate the individual contributions of our core modules and the reward function design.

As shown in Table I, removing the group-relative trajectory alignment causes a severe performance drop (30% SR, 2.1 collisions). Similarly, omitting the Obstacle Perception

Grid (OPG) yields a 40% SR and 1.6 collisions. This indicates that both components are effective. The OPG provides explicit obstacle perception to enhance risk awareness, and the alignment strategy encourages the policy to steer the generative distribution away from hazards.

We further analyze the contribution of each reward component in the proposed alignment objective (Eq. (4)), as shown in Table II. The initial policy refers to the behavior cloning model introduced in subsection III-B.

Starting from this initial policy (30.0% SR and 2.1 collisions per run), incorporating the open-space reward improves the success rate to 40.0% while reducing collisions to 1.8. This indicates that encouraging motion toward more open and traversable regions already provides more reliable navigation behavior. Adding the safety penalty further improves the policy, achieving a higher success rate of 50.0% and reducing collisions to 1.4. This demonstrates that explicitly penalizing obstacle proximity helps the policy maintain safer clearances during navigation. Combining both rewards yields the best performance, reaching 70.0% SR with only 1.2 collisions per run.

We visualize the relative safety weights assigned to candidate trajectories during policy fine-tuning, as detailed in Section III-C. As shown in Fig. 4, the policy samples multiple candidate trajectories for the same observation during the *group-relative trajectory alignment* stage. The proposed safety-oriented reward assigns higher weights to trajectories that maintain larger obstacle clearance and pass through open space, thereby prioritizing safer behaviors in the optimization and steering the policy update toward risk-aware actions.

c) Qualitative Trajectory Preference Analysis: To clarify how different policies behave during real execution, we next qualitatively compare the trajectories produced by NoMaD, FlowNav, and RaFM-Nav in representative spatial layouts. The comparison focuses on two preference factors: how much clearance each policy keeps from obstacles, and how it chooses a route when faced with spatial constraints. We evaluate three challenging spatial configurations illustrated in Fig. 5: dense clutter, narrow corridors, and multiple topological choices.

In the dense clutter environment (Fig. 5, Top), NoMaD and FlowNav generate unsafe paths that run into nearby obstacles, ending in collisions. In contrast, RaFM-Nav maintains a continuous, collision-free path through the clutter.

In the narrow corridor environment (Fig. 5, Middle), NoMaD and FlowNav move very close to the corridor boundary, leaving minimal lateral clearance (0.03 m for NoMaD and 0.08 m for FlowNav). This provides an insufficient safety buffer and inevitably leads to collisions. RaFM-Nav, however, centers the robot’s trajectory, maintaining a much safer clearance of 0.24 m to ensure a successful pass.

Finally, when presented with multiple topological choices (Fig. 5, Bottom), the environment offers an open space route and a narrow constrained route. While NoMaD and FlowNav blindly favor the narrow passage and ultimately collide, RaFM-Nav consistently selects the wider, open space route to pass safely.

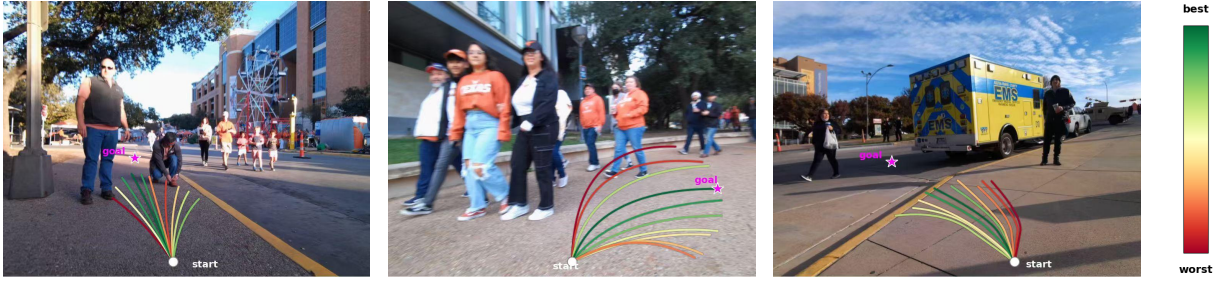


Fig. 4: Visualization of candidate trajectories and their relative safety weights computed using Eq. (9) across three representative scenes (e.g., straight, left, and right turns). For each scene, we sample multiple plausible trajectory candidates conditioned on the current observation and the goal. Trajectories are color-coded by safety weight, with green indicating the safest and red the least safe (see color bar).

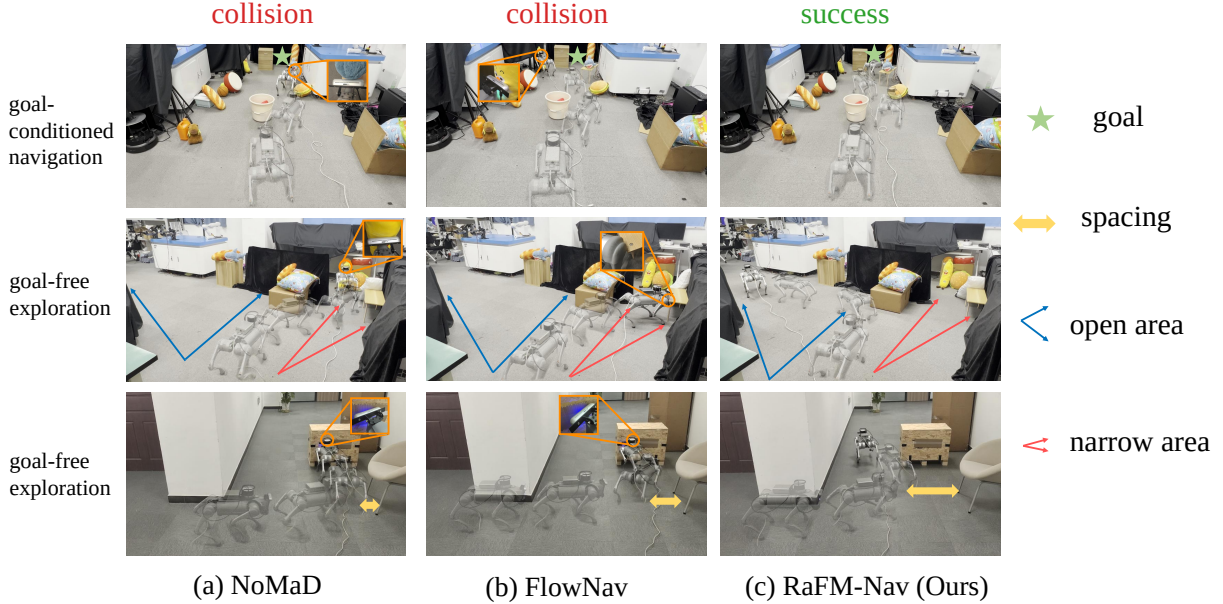


Fig. 5: Qualitative comparison in three challenging scenarios. **Top: Dense clutter.** Previous generative policies produce unsafe trajectories and collide with obstacles, while RaFM-Nav maintains collision-free navigation. **Middle: Narrow corridor.** NoMaD and FlowNav skim boundaries with low clearance; RaFM-Nav centers its trajectory for safer margins. **Bottom: Multiple topological choices.** NoMaD and FlowNav select a constrained route, whereas RaFM-Nav consistently selects wider collision-free regions.

Overall, the examples in Fig. 5 show that RaFM-Nav’s safety modeling encourages trajectories with larger clearances and more reliable route choices. Without group-relative trajectory alignment, the baselines exhibit higher variance in route selection and frequently traverse tighter regions, resulting in higher collision rates in cluttered scenarios.

d) Inference Time Analysis: We compare the inference latency across all methods with $N_{\text{step}} = 10$ (Table I). Despite a small increase in inference latency, RaFM-Nav achieves substantially better navigation performance. RaFM-Nav runs at 65.0 ms per forward pass, which is slightly slower than FlowNav (63.8 ms) and NoMaD (62.1 ms), while improving success rate from 30% to 70% and reducing collisions from 2.5 to 1.1. This marginal latency increase is entirely due to feature extraction in the Obstacle Perception Grid (OPG), and the group-relative trajectory alignment is executed fully

offline and incurs zero additional cost at inference time.

V. CONCLUSION

In this work, we presented RaFM-Nav, a framework for enhancing the safety of flow-matching visual navigation policies in densely cluttered environments. RaFM-Nav integrates a perception-level Obstacle Perception Grid to enhance the policy’s awareness of nearby risks, and introduces a group-relative trajectory alignment that guides training toward lower-risk trajectory distributions. Real-world experiments show that RaFM-Nav produces a robust navigation policy that can traverse complex, unstructured terrain while maintaining larger safety margins. Future work will extend the framework to dynamic obstacles and moving agents, and further study the scalability and sample efficiency of group-relative optimization in long-horizon navigation.

REFERENCES

- [1] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A platform for embodied AI research," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.
- [2] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, 2016.
- [3] J. Duan, S. Yu, H. L. Tan, H. Zhu, and C. Tan, "A survey of embodied AI: From simulators to research tasks," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2022.
- [4] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [5] D. S. Chaplot, D. P. Gandhi, A. Gupta, and R. R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.
- [6] J. Krantz, E. Wijmans, A. Majumdar, D. Batra, and S. Lee, "Beyond the nav-graph: Vision-and-language navigation in continuous environments," in *European Conference on Computer Vision (ECCV)*, 2020.
- [7] X. Ye and Y. Yang, "From seeing to moving: A survey on learning for visual indoor navigation (VIN)," *arXiv preprint arXiv:2002.11310*, 2020.
- [8] A. Sridhar, D. Shah, C. Glossop, and S. Levine, "NoMaD: Goal masked diffusion policies for navigation and exploration," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.
- [9] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.
- [10] 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 (ICLR)*, 2020.
- [11] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, 2024.
- [12] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations (ICLR)*, 2021.
- [13] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," in *International Conference on Learning Representations (ICLR)*, 2023.
- [14] S. Gode, A. Nayak, D. N. Oliveira, M. Krawez, C. Schmid, and W. Burgard, "FlowNav: Combining flow matching and depth priors for efficient navigation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.
- [15] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu *et al.*, "DeepSeekMath: Pushing the limits of mathematical reasoning in open language models," *arXiv preprint arXiv:2402.03300*, 2024.
- [16] M. Tranzatto, T. Miki, M. Dharmadhikari *et al.*, "Cerberus in the DARPA subterranean challenge," *Science Robotics*, 2022.
- [17] G. Georgakis, B. Bucher, K. Schmeckpeper, S. Singh, and K. Daniilidis, "Learning to map for active semantic goal navigation," in *International Conference on Learning Representations (ICLR)*, 2022.
- [18] D. Shah, B. Eysenbach, G. Kahn, N. Rhinehart, and S. Levine, "ViNG: Learning open-world navigation with visual goals," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.
- [19] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine, "GNM: A general navigation model to drive any robot," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023.
- [20] D. Shah, A. Sridhar, N. Dashora *et al.*, "ViNT: A large-scale, multi-task visual navigation backbone with cross-robot generalization," in *Conference on Robot Learning (CoRL)*, 2023.
- [21] M. Janner, Y. Du, J. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," in *International Conference on Machine Learning (ICML)*, 2022.
- [22] M. S. Albergo and E. Vanden-Eijnden, "Building normalizing flows with stochastic interpolants," in *International Conference on Learning Representations (ICLR)*, 2023.
- [23] A. Tong, N. Malkin, G. Huguet *et al.*, "Improving and generalizing flow-based generative models with minibatch optimal transport," *Transactions on Machine Learning Research*, 2024.
- [24] L. Zhang, Z. Jiang, and C. Chen, "SaferPath: Hierarchical visual navigation with learned guidance and safety-constrained control," *arXiv preprint arXiv:2603.01898*, 2026.
- [25] R. Zhang, J. Hou, C. Cheng, Q. Chen, T. Wang, and W. Zhao, "Self-imitated diffusion policy for efficient and robust visual navigation," *arXiv preprint arXiv:2601.22965*, 2026.
- [26] R. F. Prudencio, M. R. Maximo, and E. L. Colombari, "A survey on offline reinforcement learning: Taxonomy, review, and open problems," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.
- [27] C. Tang, B. Abbatematteo, J. Hu, R. Chandra, R. Martín-Martín, and P. Stone, "Deep reinforcement learning for robotics: A survey of real-world successes," *Annual Review of Control, Robotics, and Autonomous Systems*, 2025.
- [28] R. Rafailov, K. B. Hatch, A. Singh, A. Kumar, L. Smith, I. Kostrikov, P. Hansen-Estruch, V. Kolev, P. J. Ball, J. Wu, S. Levine, and C. Finn, "D5RL: Diverse datasets for data-driven deep reinforcement learning," in *Reinforcement Learning Conference (RLC)*, 2024.
- [29] M. G. Azar, Z. D. Guo, B. Piot, R. Munos, M. Rowland, M. Valko, and D. Calandriello, "A general theoretical paradigm to understand learning from human preferences," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2024.
- [30] K. Ethayarajh, W. Xu, N. Muennighoff, D. Jurafsky, and D. Kiela, "Model alignment as prospect theoretic optimization," in *International Conference on Machine Learning (ICML)*, 2024.
- [31] P. Florence, C. Lynch, A. Zeng *et al.*, "Implicit behavioral cloning," in *Conference on Robot Learning (CoRL)*, 2021.
- [32] W. Xia, Y. Yang, H. Wu, X. Ma, T. Kong, and D. Hu, "Human-assisted robotic policy refinement via action preference optimization," *arXiv preprint arXiv:2506.07127*, 2025.
- [33] L. Shao, T. Migimatsu, Q. Zhang, K. Yang, and J. Bohg, "Concept2Robot: Learning manipulation concepts from instructions and human demonstrations," *The International Journal of Robotics Research*, 2021.
- [34] J. Hejna and D. Sadigh, "Inverse preference learning: Preference-based RL without a reward function," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.
- [35] DeepSeek-AI, D. Guo, D. Yang, H. Zhang, J. Song, P. Wang, Q. Zhu, R. Xu, R. Zhang, S. Ma *et al.*, "DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning," *Nature*, 2025.
- [36] D. Kim, S. Park, H. Jang, J. Shin, J. Kim, and Y. Seo, "Robot-R1: Reinforcement learning for enhanced embodied reasoning in robotics," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.
- [37] N. Hirose, A. Sadeghian, M. Vázquez, P. Goebel, and S. Savarese, "GoNet: A semi-supervised deep learning approach for traversability estimation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.
- [38] D. Shah, B. Eysenbach, N. Rhinehart, and S. Levine, "Rapid exploration for open-world navigation with latent goal models," in *Conference on Robot Learning (CoRL)*, 2022.
- [39] S. Triest, M. Sivaprakasam, S. J. Wang, W. Wang, A. M. Johnson, and S. Scherer, "TartanDrive: A large-scale dataset for learning off-road dynamics models," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022.
- [40] N. Hirose, D. Shah, A. Sridhar, and S. Levine, "SACSoN: Scalable autonomous control for social navigation," *IEEE Robotics and Automation Letters*, 2024.
- [41] 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.
- [42] D. M. Nguyen, M. Nazeri, A. Payandeh, A. Datar, and X. Xiao, "Toward human-like social robot navigation: A large-scale, multi-modal, social human navigation dataset," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.
- [43] J. Liang, D. Das, D. Song, M. N. H. Shuvo, M. Durrani, K. Tarantath, I. Penskiy, D. Manocha, and X. Xiao, "GND: Global navigation dataset with multi-modal perception and multi-category traversability in outdoor campus environments," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2025.