2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

DreamFlow: Local Navigation Beyond Observation via Conditional Flow Matching in the Latent Space

Jiwon Park[1†], Dongkyu Lee[2,3†], I Made Aswin Nahrendra[2,4], Jaeyoung Lim[5], and Hyun Myung[2*], *Senior Member, IEEE*

***Abstract*— Local navigation in cluttered environments often suffers from dense obstacles and frequent local minima. Conventional local planners rely on heuristics and are prone to failure, while deep reinforcement learning (DRL)-based approaches provide adaptability but are constrained by limited onboard sensing. These limitations lead to navigation failures because the robot cannot perceive structures outside its field of view. In this paper, we propose DreamFlow, a DRL-based local navigation framework that extends the robot's perceptual horizon through conditional flow matching (CFM). The proposed CFM-based prediction module learns probabilistic mapping between local height map latent representation and broader spatial representation conditioned on navigation context. This enables the navigation policy to predict unobserved environmental features and proactively avoid potential local minima. Experimental results demonstrate that DreamFlow outperforms existing methods in terms of latent prediction accuracy and navigation performance in simulation. The proposed method was further validated in cluttered real-world environments with a quadrupedal robot. The project page is available at `https://dreamflow-icra.github.io`.**

I. Introduction

Safe navigation in cluttered environments is a key requirement for autonomous mobile systems, including search and rescue [1], exploration [2], and environmental monitoring [3]. Such environments contain dense obstacles and severe visual occlusions, which make reaching target locations highly challenging and call for structured navigation strategies. To address this, hierarchical navigation pipelines [4], [5] have been widely adopted, decomposing navigation into perception, planning, and control. Among them, the role of the planning module is to generate feasible commands that guide the robot toward its goal, based on the environmental representation constructed by the perception module.

†These authors contributed equally.
*Corresponding author: Hyun Myung.
[1]Robotics Program, KAIST (Korea Advanced Institute of Science and Technology), Daejeon 34141, South Korea. ziwon@kaist.ac.kr
[2]School of Electrical Engineering, KAIST (Korea Advanced Institute of Science and Technology), Daejeon 34141, South Korea. {dklee, anahrendra, hmyung}@kaist.ac.kr
[3]URobotics, Seoul, South Korea. dklee@urobotics.ai
[4]KRAFTON, Seoul, South Korea. anahrendra@krafton.com
[5]Dept. of Electrical Engineering and Computer Science, UC Berkeley, USA. jaeyounglim@berkeley.edu

This work was supported in part by the Technology Commercialization support program, through the Korea Innovation Foundation funded by the Ministry of Science and ICT, and in part by Korea Evaluation Institute of Industrial Technology (KEIT) funded by the Korea Government (MOTIE) under Grant 20018216, Development of mobile intelligence SW for autonomous navigation of legged robots in dynamic and atypical environments for real application. The students are supported by BK21 FOUR.

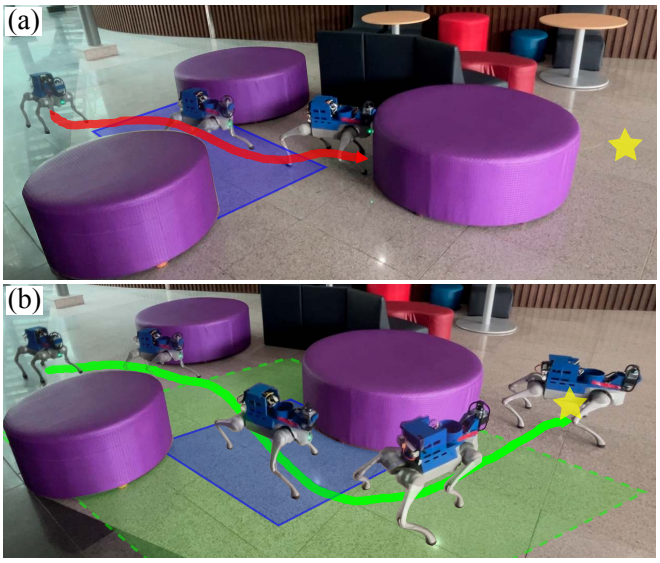


Fig. 1. An illustration of local navigation to the goal point (yellow star) in a cluttered environment with obstacles. (a) Local navigation with only a limited onboard sensing range (blue boxed region) may lead to local minima when obstacles lie beyond the sensor range. (b) DreamFlow enables the robot to find optimal actions within the observed sensing range (blue boxed region) by leveraging conditional flow matching (CFM)-based prediction of extended latent representations (green boxed region). The red and green lines indicate the executed trajectories, respectively.

The planning module is typically divided into global planners [6]–[8], which compute long-horizon paths for high-level guidance, and local planners [9]–[11], which operate over shorter horizons to react to immediate disturbances and obstacles. However, the absence of reliable maps in real-world scenarios makes global planning impractical, especially in previously unseen or rapidly changing environments. In such cases, navigation often relies on the local navigation, which emphasizes reactive obstacle avoidance and short-horizon goal reaching under limited visibility.

A key challenge in local navigation is the *local minima* problem, a situation where the robot becomes trapped because no available action leads closer to the goal. Typical cases include U-shaped obstacles or dead-end corridors, where progressing forward risks a collision while retreating increases the distance to the target. Such situations arise because the local planner considers only the robot's immediate perceptual horizon, prioritizing short-term safety over long-term goal progress. As a result, the robot may misjudge the goal as unreachable even though a feasible path exists beyond its field of view. Conventional local planners rely on simple heuristics [12]–[14] to escape such traps, but frequently fail

in cluttered or highly occluded environments.

Learning-based methods [15]–[19] have recently emerged as a promising alternative to local navigation, with deep reinforcement learning (DRL) receiving particular attention for its ability to train policies directly through interaction with the environment while incorporating robot kino-dynamics and feedback signals. These approaches reduce reliance on manually designed heuristics and enable more adaptive behaviors in cluttered environments. However, a DRL policy still relies on a partially observable map acquired through an onboard perception system. This limited perception often prevents the robot from reasoning beyond its immediate view, leaving it vulnerable to local minima in complex environments. Although some recent works have improved navigation performance by incorporating robot dynamics over time [20] and by reconstructing more complete environment states from partial observations [21], [22], they still rely on planning within limited local areas.

In this context, we propose DreamFlow, a DRL-based local navigation framework that extends the implicit perceptual range by leveraging the emerging generative model, conditional flow matching (CFM) [23]–[26], as illustrated in Fig. 1. The proposed CFM-based prediction module captures the inherent uncertainty of unobserved regions by learning probabilistic mappings between local and global representations, conditioned on navigation context. Specifically, the network learns optimal transport flows that map latent representations from local height maps to those of wider regions. Then, we integrate the proposed CFM prediction module into the DRL-based navigation pipeline adapted from [21]. Therefore, the policy reasons about unobserved environmental features and proactively avoids potential local minima.

We evaluated our proposed method in both simulation and real-world experiments. The results demonstrate that local navigation policies leveraging prediction of extended latent representations achieve improved navigation performance and robustness, particularly in complex environments prone to local minima. In summary, the main contributions of this paper are as follows:

- We introduce DreamFlow, a DRL-based local navigation framework that addresses the local minima problem arising from the limited range of onboard sensors.
- We propose a latent prediction module leveraging CFM to predict extended environmental representations from local height maps under contextual conditions.
- We demonstrate the effectiveness of DreamFlow, which outperforms existing methods in simulation and is further validated in real-world environments with cluttered obstacles and local minima.

II. Related Works

A. Local Navigation

The aim of local navigation is to guide robots safely around obstacles and toward designated goals, using only local perception. Classical reactive methods [12]–[14], [27] provide computationally efficient solutions but are prone to myopic decisions, often leading to suboptimal paths in cluttered environments.

Recently, learning-based approaches have been proposed to address the local minima problem. Some research [16], [28] leverages temporal information from past interactions, enabling robots to recall failed attempts and avoid repetitive behaviors, while other works [14], [29] integrate global planning guidance into local navigation through cost map potentials or intermediate waypoints. However, temporal methods struggle against novel obstacle configurations unseen during training, and hybrid approaches rely on simplified environment representations that reduce the local planner's ability to cope with obstacles beyond the sensor range. In this context, our approach focuses on enabling the local planner to directly predict environmental structure beyond the immediate sensor range, proactively avoiding potential traps without requiring global maps or temporal memory.

B. Flow Matching in Robotics

Flow matching (FM) is a simulation-free generative modeling framework that learns continuous normalizing flows (CNF) via neural ordinary differential equations (ODE) to transport probability distributions from source to target spaces [23], [30]. Unlike diffusion models [31], which rely on iterative denoising processes that are computationally expensive during both training and sampling, FM offers a more direct approach by learning deterministic velocity fields to guide the transformation between distributions. CFM [26] extends FM by conditioning on source–target pairs, thereby enabling the construction of more flexible transport maps.

Recent robotics applications of CFM have targeted trajectory-level planning and control, including behavioral cloning for motion field learning [32], manipulation policy generation from 3D observations [33], [34], and depth-prior-based mobile robot navigation [35]. Although these works successfully demonstrate CFM's effectiveness in direct policy learning by predicting future robot actions, there remains an opportunity to investigate how CFM can enhance spatial representations for improved robot navigation, which is the main focus of this work. Although these works successfully demonstrate CFM's effectiveness in direct policy learning by predicting future robot actions, there remains an opportunity to investigate how CFM can enhance spatial representations for improved robot navigation, which is the main focus of this work.

III. DreamFlow

A. Overall Architecture

The objective of DreamFlow is to enable robots to reach target positions while avoiding redundant exploration or getting stuck in local minima. We formulate this problem as a DRL-based navigation task, building upon the approach of Zhang *et al.* [21]. The overall navigation architecture, illustrated in Fig. 2, is designed as an asymmetric actor-critic framework [36]. The navigation policy π_{nav} outputs the robot's body velocity actions $\mathbf{a}_t$ based on the observations,

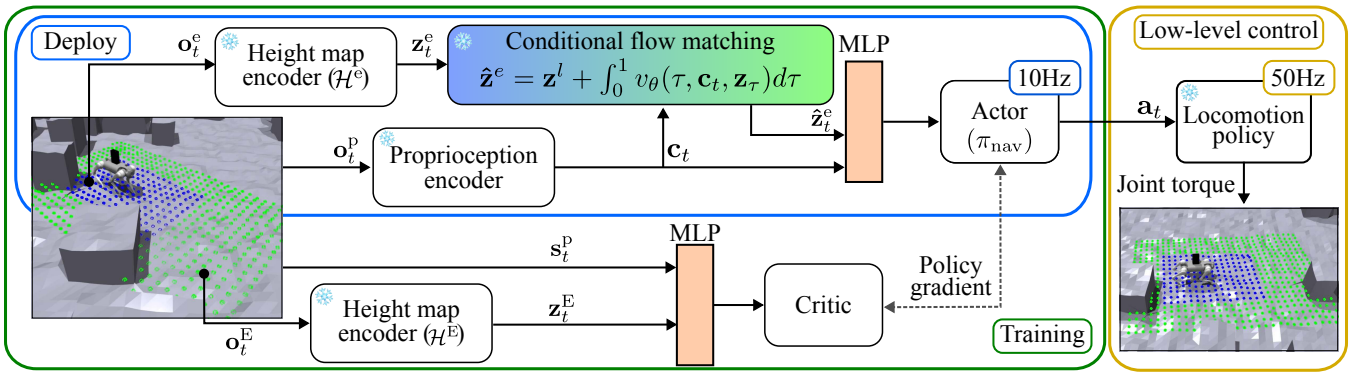


Fig. 2. The overall framework of DreamFlow. During deployment, the local height map $\mathbf{o}_t^{\mathrm{e}}$ (blue dots, indicating observable terrain) is encoded into the local latent vector $\mathbf{z}_t^{\mathrm{e}}$. Subsequently, the pre-trained CFM predicts the extended latent vector $\hat{\mathbf{z}}_t^{\mathrm{e}}$ conditioned on the context vector $\mathbf{c}_t$ encoded from the proprioceptive observations $\mathbf{o}_t^{\mathrm{p}}$. The predicted latent vector $\hat{\mathbf{z}}_t^{\mathrm{e}}$ represents information from the extended height map region (green dots, indicating terrain beyond sensor range). The navigation policy π_{nav} takes the concatenated $\mathbf{c}_t$ and $\hat{\mathbf{z}}_t^{\mathrm{e}}$ as input to produce the high-level velocity action $\mathbf{a}_t$. The locomotion policy then generates the low-level joint actions to control the robot. When training the navigation policy, the privileged states $\mathbf{s}_t^{\mathrm{p}}$ and the extended latent vector $\mathbf{z}_t^{\mathrm{E}}$ from the extended height map $\mathbf{o}_t^{\mathrm{E}}$ are used to train the critic network.

while a pretrained locomotion policy [37] is employed as a low-level controller. Both the actor and critic networks are trained using the proximal policy optimization (PPO) algorithm [38].

A notable feature of DreamFlow is its use of asymmetric exteroception height map observations between the actor and critic. The actor uses a local height map $\mathbf{o}_t^{\mathrm{e}} \in \mathbb{R}^{H\times W}$, which the robot can actually observe, to derive a local environmental latent representation. In contrast, the critic utilizes an extended privileged height map $\mathbf{o}_t^{\mathrm{E}} \in \mathbb{R}^{H'\times W'}$, with a larger sensing range beyond the onboard sensors, to extract an extended latent representation. Here, H, W, H', and W' denote the height and width of the local and extended height maps, respectively, subject to the conditions $H < H'$ and $W < W'$. The proprioceptive observations $\mathbf{o}_t^{\mathrm{p}}$ and privileged states $\mathbf{s}_t^{\mathrm{p}}$ contain navigation-related information such as the target position, heading command, and previous actions, along with measurements from the inertial measurement unit (IMU) and joint states.

Compared with Zhang *et al.* [21], the observation encoding is simplified, as our contribution mainly lies in the latent prediction module. Two pre-trained variational autoencoders (VAEs) are utilized for height map encoders, $\mathcal{H}^{\mathrm{e}}$ and $\mathcal{H}^{\mathrm{E}}$, to extract latent representation of local and extended height map, respectively. For proprioceptive inputs, we adopt context-aided estimator network (CENet) [37]. Subsequently, the velocity field trained via CFM, detailed in Section III-B, predicts the extended latent vector $\hat{\mathbf{z}}_t^{\mathrm{e}}$ by transporting the local latent vector $\mathbf{z}_t^{\mathrm{e}}$ toward the extended latent vector $\mathbf{z}_t^{\mathrm{E}}$. Finally, the predicted latent representation vector $\hat{\mathbf{z}}_t^{\mathrm{e}}$ and the proprioceptive features are concatenated and fed into the actor network.

B. Conditional Flow Matching in Latent Space

To overcome the limitations of partial observability in local navigation, we propose an implicit environment prediction module using latent-to-latent CFM [25]. The proposed CFM-based module predicts the extended latent space from the local latent space, enabling the navigation policy to make more informed decisions.

We formulate this task as the problem of learning a continuous flow that maps from the local environmental latent vector $\mathbf{z}_t^{\mathrm{e}} \in \mathbb{R}^d$ to the extended environmental latent vector $\mathbf{z}_t^{\mathrm{E}} \in \mathbb{R}^d$, where d is the dimensionality of the latent space. The $\mathbf{z}_t^{\mathrm{e}}$ is an embedded vector from the limited local height map within the robot's perception range, whereas the $\mathbf{z}_t^{\mathrm{E}}$ embeds a larger spatial structure beyond the current observed height map. Therefore, the deterministic flow, also referred to as a velocity field $v_\theta(\tau, \mathbf{z})$, is defined by the ODE as follows:

$$\frac{d\mathbf{z}}{d\tau} = v_\theta(\tau, \mathbf{z}), \tag{1}$$

where $\tau \in [0, 1]$ is the step size for interpolation between the source and target distributions, and $\mathbf{z}$ denotes the latent representation vector. According to prior works on FM [23], [26], the target velocity field is intractable because it involves integrating over all pairs sampled from the Gaussian noise distribution and mapped to the target data distribution. However, in DreamFlow, $v_\theta(\tau, \mathbf{z})$ becomes learnable because we sample the initial latent $\mathbf{z}_0$ directly from the local latent distribution p_0, and the target latent $\mathbf{z}_1$ from the extended latent distribution $q(\mathbf{z})$.

Nonetheless, the prediction of the extended latent is inherently multi-modal, as multiple extended environments might correspond to a single local observation. To address this, we incorporate a condition vector $\mathbf{c}_t$ into (1) for deterministic context-aware mapping as follows:

$$\frac{d\mathbf{z}}{d\tau} = v_\theta(\tau, \mathbf{c}_t, \mathbf{z}), \tag{2}$$

where $v_\theta(\tau, \mathbf{c}_t, \mathbf{z})$ is the conditioned velocity field, and the $\mathbf{c}_t$ is the conditioning context that encodes the robot's state and goal information at each time step. The solution of (2) provides a trajectory that connects the initial local latent representation $\mathbf{z}_0 = \mathbf{z}_t^{\mathrm{e}}$ to the target extended representation $\mathbf{z}_1 = \mathbf{z}_t^{\mathrm{E}}$. In this way, the flow learns the optimal transport from an initial distribution of local latent representations $p_0(\mathbf{c}_t)$ to the target distribution of extended latent representations $q(\mathbf{z}|\mathbf{c}_t)$.

The conditioned velocity field $v_\theta(\tau, \mathbf{c}_t, \mathbf{z})$ is optimized by minimizing the following loss function:

$$\mathcal{L}(\theta) = \mathbb{E}_{\tau,\, \mathbf{z}_0 \sim p_0(\mathbf{c}_t),\, \mathbf{z}_1 \sim q(\mathbf{z}|\mathbf{c}_t)} \left[\| v_\theta(\tau, \mathbf{c}_t, \mathbf{z}) - \mathbf{u}_\tau(\mathbf{z}|\mathbf{c}_t) \|^2 \right], \tag{3}$$

where $\mathbf{u}_\tau(\mathbf{z}|\mathbf{c}_t) = \mathbf{z}_1 - \mathbf{z}_0$ is the target velocity field under optimal transport, and τ is sampled from the uniform distribution $\mathcal{U}[0, 1]$. Therefore, $\mathbf{z}_\tau$ can be obtained by linear interpolation between the initial and target latent representations as follows:

$$\mathbf{z}_\tau = (1 - \tau)\mathbf{z}_0 + \tau \mathbf{z}_1. \tag{4}$$

C. DreamFlow Training

The DreamFlow training is conducted in two stages. First, CFM is trained to infer the prediction of the extended latent representation from the local observation latent representation. Then, the navigation policy π_{nav} is trained to output velocity commands based on the latent representation predicted by the frozen CFM model.

1) CFM Training: The objective of the CFM training is to learn the velocity field network that transports the local latent distribution $p_0(\mathbf{c}_t)$ to the extended latent distribution $q(\mathbf{z}|\mathbf{c}_t)$. We first pre-train two navigation policies using local and extended height maps, respectively. Then, we collect a dataset $\mathcal{D} = \{(\mathbf{z}_i^l, \mathbf{z}_i^e, \mathbf{c}_i)\}_{i=1}^N$ by gathering latent pairs and conditions during multi-agent navigation episodes in the IsaacGym [39] simulation. Formally, the loss function in (3) is formulated as follows:

$$\mathcal{L}_{\text{DreamFlow}}(\theta) = \mathbb{E}_{\tau \sim \mathcal{U}[0,1],\, (\mathbf{z}_t^{\text{e}}, \mathbf{z}_t^{\text{E}}, \mathbf{c}_t) \sim \mathcal{D}} \left[\| v_\theta(\tau, \mathbf{c}_t, \mathbf{z}_\tau) - (\mathbf{z}_t^{\text{E}} - \mathbf{z}_t^{\text{e}}) \|^2 \right], \tag{5}$$

where $\mathbf{z}_\tau = (1 - \tau)\mathbf{z}_t^{\text{e}} + \tau \mathbf{z}_t^{\text{E}}$. Therefore, $v_\theta(\tau, \mathbf{c}_t, \mathbf{z}_\tau)$ is optimized to transport local latent vector $\mathbf{z}_t^{\text{e}}$ to extended latent vector $\mathbf{z}_t^{\text{E}}$ conditioned on $\mathbf{c}_t$, as depicted in Fig. 3.

2) Navigation Policy Training: During the navigation policy training, we use the frozen CFM to predict extended latent representations. The initial latent is set to the local latent representation $\mathbf{z}_t^{\text{e}}$, and then the predicted extended latent representation $\hat{\mathbf{z}}_t^{\text{e}}$ is generated using the CFM velocity field $v_\theta(\tau, \mathbf{c}_t, \mathbf{z}_\tau)$ as follows:

$$\hat{\mathbf{z}}_t^{\text{e}} = \mathbf{z}_t^{\text{e}} + \int_0^1 v_\theta(\tau, \mathbf{c}_t, \mathbf{z}_\tau) d\tau. \tag{6}$$

In the deployment of (6), we use Euler integration with a discrete step size $\tau_k = \frac{1}{K}$:

$$\hat{\mathbf{z}}_t^{\text{e}} \approx \mathbf{z}_t^{\text{e}} + \sum_{k=0}^{K-1} \tau_k v_\theta(\tau_k, \mathbf{c}_t, \mathbf{z}_{\tau_k}), \tag{7}$$

where K is the number of integration steps, $k \in \{0, 1, ..., K-1\}$ indexes the integration steps, and the final $\hat{\mathbf{z}}_t^{\text{e}}$ approximates the integral result. The navigation policy π_{nav} is then trained using the predicted extended latent $\hat{\mathbf{z}}_t^{\text{e}}$, while keeping the remaining training setup consistent with the Zhang *et al.* [21]. This enables the robot to exploit extended spatial representations while operating with only local sensor observations.

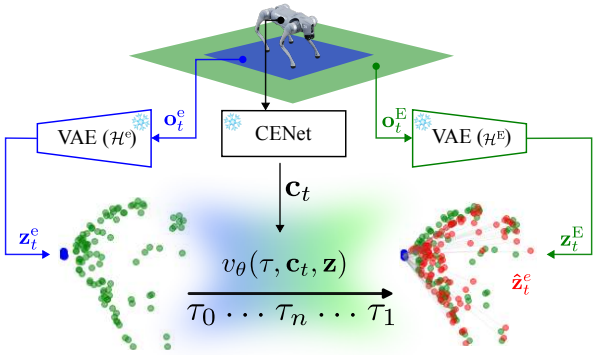


Fig. 3. The CFM training pipeline. The training dataset is collected using pre-trained height map encoders ($\mathcal{H}^{\text{e}}$ and $\mathcal{H}^{\text{E}}$) and a proprioceptive encoder (CENet). A local height map $\mathbf{o}_t^{\text{e}}$ (blue boxed region) is encoded into a local latent representation $\mathbf{z}_t^{\text{e}}$, and a privileged extended height map $\mathbf{o}_t^{\text{E}}$ (green boxed region) is encoded into an extended latent representation $\mathbf{z}_t^{\text{E}}$. During training, CFM learns a velocity field $v_\theta(\tau, \mathbf{c}_t, \mathbf{z})$, conditioned on the contextual vector $\mathbf{c}_t$, that transports $\mathbf{z}_t^{\text{e}}$ at τ_0 towards at τ_1. Consequently, CFM maps $\mathbf{z}_t^{\text{e}}$ (blue dots) to the predicted latent vector $\hat{\mathbf{z}}_t^{\text{e}}$ (red dots), which aligns with $\mathbf{z}_t^{\text{E}}$ (greed dots) in the latent space.

IV. Experiments

We performed both simulation and real-world experiments to validate the effectiveness of the proposed method. From a fundamental perspective, our experiments were designed to address the following questions:

Q1: Can CFM effectively predict an extended latent vector from a partially observed local latent vector? We evaluated our CFM prediction module by measuring how accurately it predicts the target latent vector $\mathbf{z}_t^{\text{E}}$ from the local latent vector $\mathbf{z}_t^{\text{e}}$ compared with other model architectures (Section V-A).

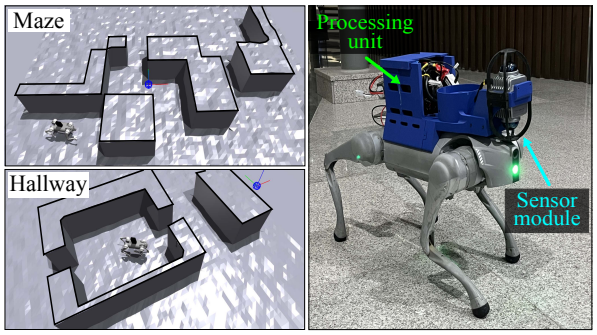
Q2: Do the latent predictions improve the performance of local navigation? We empirically validated the assumption that the predicted latent vector $\hat{\mathbf{z}}_t^{\text{e}}$ contains useful information about unobserved terrain through navigation experiments in simulation environments (Section V-B).

Q3: Is the proposed framework effective in real-world scenarios? We deployed DreamFlow in real-world environments, validating its generalization capability when exposed to real-world uncertainties such as sensor noise and computational constraints (Section V-C).

A. Implementation Details

The simulation environment for training and evaluation was built using IsaacGym [39]. During training, obstacles of varying sizes were randomly distributed on a flat terrain of size $8 \times 8\,\text{m}$. The obstacle density was progressively increased through curriculum learning to gradually enhance the environmental complexity. The local height map covered an area of $3 \times 2\,\text{m}$, while the privileged extended height map size was set to $6 \times 4\,\text{m}$. Both height maps were discretized with a spatial resolution of $0.1\,\text{m}$.

The proposed CFM model was trained on a dataset consisting of 4M samples, with two disjoint test sets of 64K samples each for the in-distribution (ID) and out-of-distribution (OOD) evaluations. The ID test set comprised



(a) (b)

Fig. 4. (a) The simulation environments for evaluating navigation performance. The Maze environment consists of multiple corners leading to the goal, while the Hallway environment consists of narrow passages. (b) Our quadruped robot, used in real-world experiments, is equipped with a processing unit and a sensor module mounted on its body.

environments similar to the training distribution, while the OOD test set included environments with random pits, introducing terrain features not encountered during training. The CFM model employed a 3-layer MLP with 256 hidden units and a 32-dimensional conditioning vector. It was trained for 1,000 epochs with a batch size of 512. During inference, the integration step K for the CFM and the diffusion model was set to 10.

For training the navigation policy, the height map VAEs were implemented as 3-layer MLPs with 128-dimensional latent spaces and pre-trained for 3,000 epochs. The training took approximately 3 and 2.5 hours for the CFM and navigation policy, respectively, on an NVIDIA A5000 GPU.

B. Experimental Setup

1) Latent Prediction Evaluation: The performance of the latent prediction was evaluated using cosine similarity between the predicted and ground-truth latent representations across the ID and OOD test sets. Additionally, the inference time of the models was evaluated on an NVIDIA 3060Ti GPU to assess computational efficiency. We compared the latent predictions using MLP, Diffusion [31], FM, and CFM, which was employed in DreamFlow. All models were trained on the same dataset and training settings except for differences in the latent prediction model architecture for a fair comparison.

2) Navigation Performance in Simulation: For the evaluation of navigation performance, two environments were designed: Maze and Hallway, as illustrated in Fig. 4(a). The Maze environment is characterized by multiple corners and dead ends, requiring the robot to make strategic decisions to avoid local minima. Additionally, we set `Easy` and `Hard` scenarios according to the goal location. In the Hallway environment, a goal is located at the end of a narrow and wall-bounded passage, which is designed to evaluate the navigation capabilities of the robot in confined spaces without collisions. These three scenarios were performed for 1,000 episodes each.

We compared DreamFlow against three different methods as follows:

TABLE I. Comparison of latent prediction architectures on ID and OOD test sets. Cos. represents cosine similarity between predicted and ground-truth extended latents, where higher values indicate better prediction quality. Inference time is averaged per sample. The best results are indicated in **bold**.

Architecture	Cos. [mean ± std]		Time [ms] (↓)
	ID	OOD	
MLP	0.928 ± 0.012	0.702 ± 0.136	**0.0007**
Diffusion [31]	0.911 ± 0.349	0.919 ± 0.380	0.0218
FM	0.873 ± 0.020	0.784 ± 0.105	0.0117
CFM (Ours)	**0.980** ± 0.015	**0.973** ± 0.017	0.0122

- **Baseline**: A navigation policy trained without a latent prediction module, relying solely on the current local height map observations.
- **Zhang *et al.* [21]**: A state-of-the-art local navigation policy that utilizes LSTM to encode memories of past height map observations.
- **Diffusion [31]**: A navigation policy trained using latent predictions generated by a diffusion model.
- **DreamFlow (Ours)**: Our proposed navigation policy trained using latent predictions generated by a CFM.

We evaluated the navigation policies using three performance metrics: i) navigation success rate (SR), ii) collision rate (CR) to measure safe navigation, and iii) success weighted by path length (SPL) [40] to assess path efficiency. A trial is regarded as successful if the robot reaches the goal within $0.4\,\mathrm{m}$ in $20\,\mathrm{s}$. The CR is defined as the average number of collisions incurred by the non-foot links of the robot. The SPL is formulated as follows:

$$\mathrm{SPL} = \frac{1}{N}\sum_{i=1}^{N} S_i \cdot \frac{L_i}{\max(P_i, L_i)}, \tag{8}$$

where $S_i \in \{0, 1\}$, L_i, and P_i are the success rate, the optimal path length, and the actual path length for the i-th episode, respectively.

3) Real-World Deployment: Real-world experiments were performed on a Unitree Go2 robot, as shown in Fig. 4(b), evaluating both the baseline and our proposed method, DreamFlow. We used two Livox Mid-360 LiDARs to obtain point cloud data as the exteroception. For the perception module, we employed FAST-LIO [41] for odometry and Elevation Mapping [42] for height mapping. The perception and high-level navigation policy were deployed on an Intel NUC Core i7-11700K CPU, while the low-level controller [37] was deployed on the robot's onboard Jetson Orin NX.

V. Results and Analysis

A. Latent Prediction

Table I summarizes the latent prediction methods on both ID and OOD test sets. The deterministic MLP performs reasonably well on ID data but suffers substantial degradation on OOD samples, with sharply increased variance that reveals overfitting and unstable predictions on unfamiliar terrains.

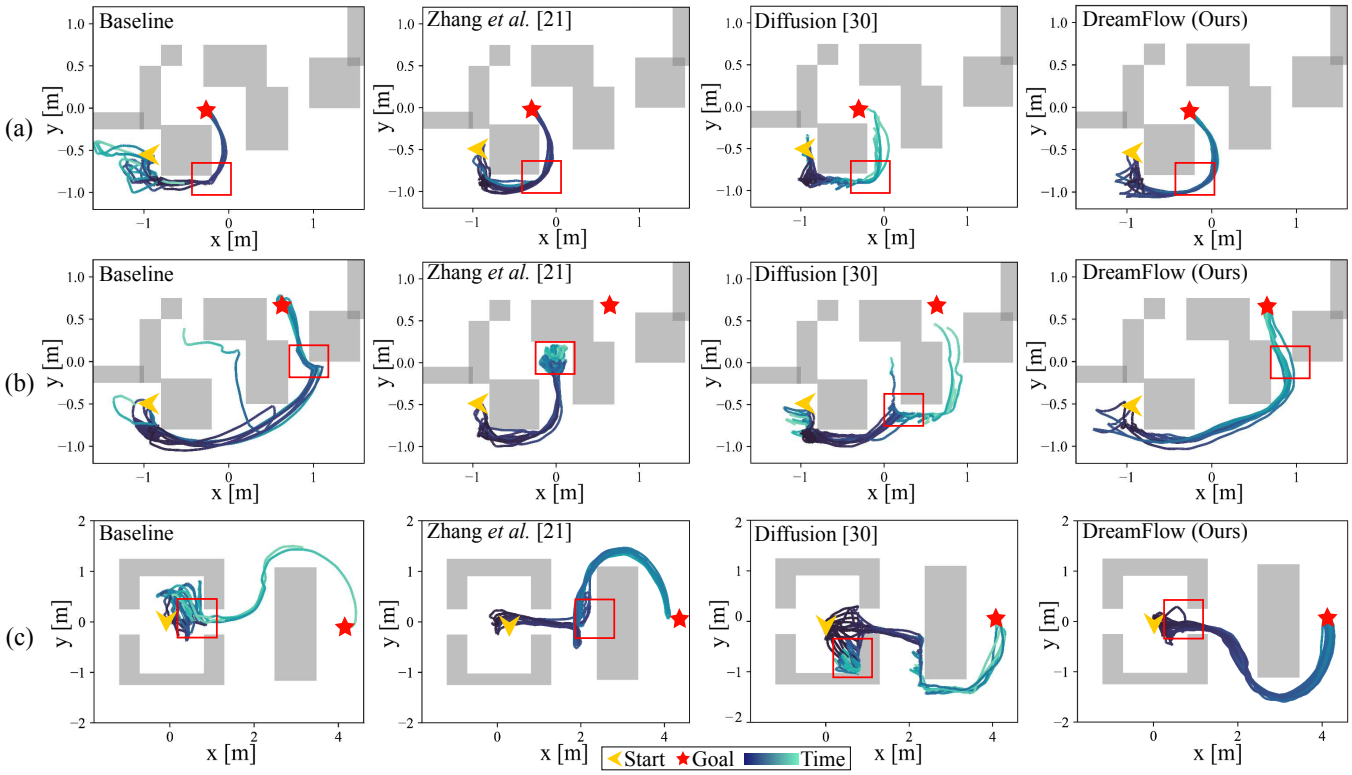


Fig. 5. Comparison of local navigation performance across three simulation scenarios: (a) Maze (`Easy`), (b) Maze (`Hard`), and (c) Hallway. Robot trajectories are depicted from a top view and overlaid on a height map, starting from initial positions (yellow markers) and ending at goal locations (red stars). Trajectories are color-coded according to time progression, with a gradient from dark blue (start) to cyan (goal). DreamFlow demonstrates smoother trajectories, with better obstacle avoidance and more efficient path selection (highlighted by the red box). In contrast, other methods show frequent obstacle contacts and often get stuck in local minima.

In contrast, flow-based methods exhibit stronger generalization, thanks to their generative capability to predict unseen regions. Although FM achieves more stable performance on OOD compared with MLP, its improvement remains limited, suggesting that flow modeling alone is insufficient without contextual guidance. Diffusion models [31], another class of generative model, demonstrate competitive performance across distributions. However, their variance remains high and the inference time is considerably longer, as the stochastic iterative sampling procedure accumulates noise across multiple steps. Consequently, they are less practical for real-time navigation.

The key advantage of CFM lies in its conditioning on the robot navigation context, which improves the mean prediction quality and substantially reduces the variance, even on OOD data. By leveraging the navigational context rather than memorizing terrain-specific patterns, CFM maintains stable predictions across unseen environments.

B. Navigation Performance

Table II and Fig. 5 summarize the navigation performance across Maze and Hallway environments of increasing difficulty. In the Maze environments, the baseline policy often relies on redundant exploration, reaching goals only after long detours and frequent obstacle contacts. Zhang *et al.* [21] and the Diffusion-based method [31] suffer from local minima in the `Hard` scenario, where short-sighted decisions trap the robot and severely reduce success rates. In contrast, DreamFlow avoids these local minima by predicting extended terrain, enabling more directed exploration and substantially higher goal-reaching success. Furthermore, DreamFlow achieves this with minimal collisions, as the predicted extended latent representation allows the policy to anticipate potential contact situations far before the robot approaches the obstacles.

The Hallway environment requires reasoning about traversability of narrow gaps relative to the robot's body dimensions. The compared policies frequently interpret passable corridors as a single large obstacle with high risk, hindering the robot from passing through. However, DreamFlow policy is able to discern this passage and safely navigates the robot through it, resulting in the highest success rate and shortest SPL.

Overall, these results validate that DreamFlow policy with the CFM's probabilistic prediction of extended terrain equips the policy with foresight that is essential for escaping local minima, reducing redundant exploration, and proactively avoiding collisions in complex environments.

C. Real World Experiments

We validated our approach through real-world experiments in two environments, as shown in Fig. 6. The *narrow* environment (rows a-b) consists of a maze-like configuration with tight passages that challenge the robot's ability to navigate through confined spaces. The *cluttered* environment (rows c-d) features two box obstacles and one wall segment that

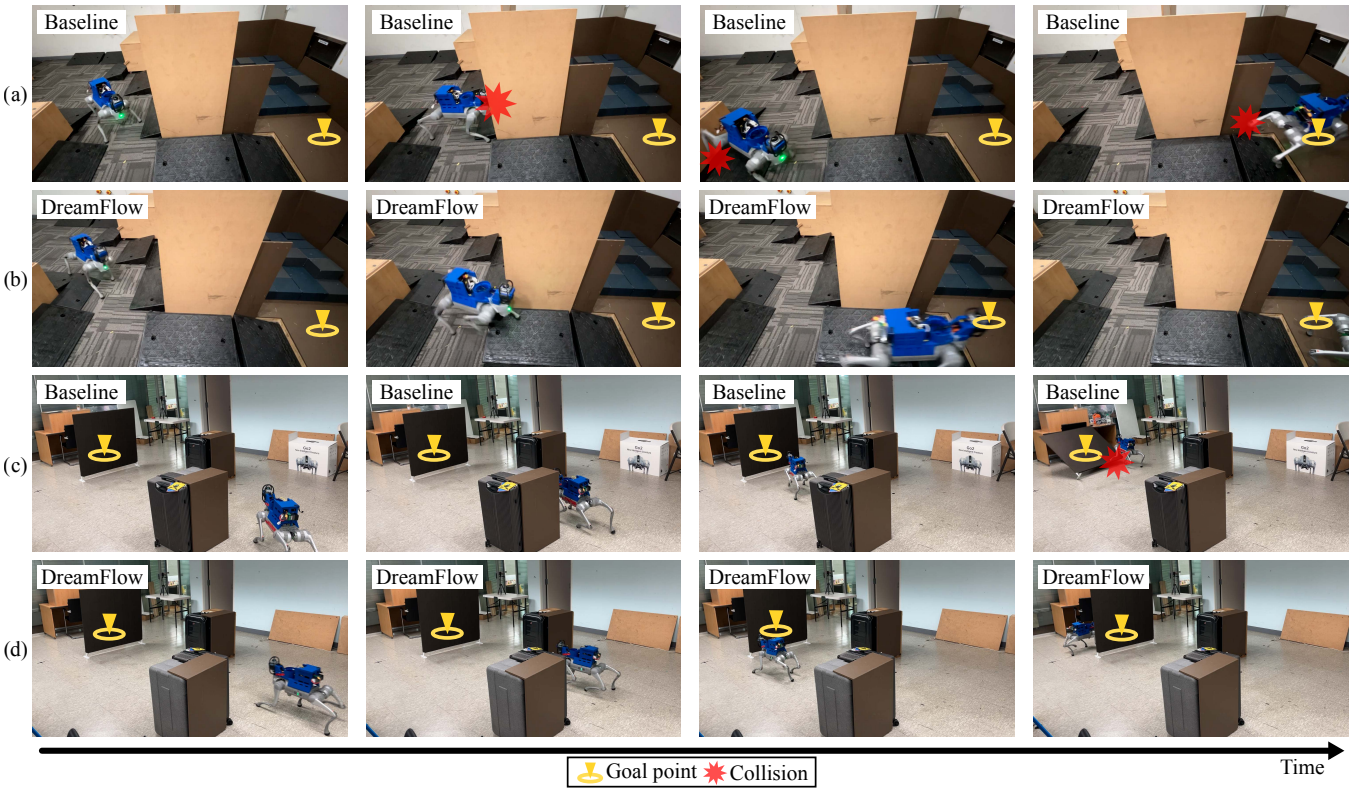


Fig. 6. Real-world navigation experiments on the quadruped robot, comparing baseline and DreamFlow. Time progresses from left to right across four snapshots. Yellow and red markers indicate goals and collision events, respectively. (a)–(b) Narrow environment with tight passages: the baseline policy frequently collides with walls at corners, whereas DreamFlow achieves collision-free navigation by anticipating corridor layouts beyond immediate perception. (c)–(d) Cluttered environment with box obstacles and a wall segment: the baseline collides with the wall due to limited perception, while DreamFlow successfully navigates without collisions through predictive terrain modeling.

TABLE II. Navigation performance in Maze and Hallway environments. Metrics are success rate (SR), success weighted by path length (SPL), and collision rate (CR). CFM achieves the highest SR and SPL while maintaining the lowest CR, demonstrating its ability to avoid local minima and collisions compared with other methods. The best results are indicated in **bold**.

Method	Maze (Easy)			Maze (Hard)			Hallway		
	SR↑	SPL↑	CR↓	SR↑	SPL↑	CR↓	SR↑	SPL↑	CR↓
Baseline	83.2	0.23	3.9	76.5	0.37	54.8	35.8	0.21	5.1
Zhang *et al.* [21]	95.3	0.33	2.5	5.4	0.03	15.6	25.1	0.12	4.9
Diffusion [31]	88.4	0.28	3.1	68.9	0.32	43.9	33.9	0.19	23.6
DreamFlow (Ours)	**99.6**	**0.35**	**0.9**	**83.1**	**0.45**	**8.9**	**89.8**	**0.58**	**2.3**

create multiple occlusions between the robot and the target position.

As illustrated in Fig. 6(a), the baseline method in the narrow environment frequently collides with walls (red markers), particularly when attempting to navigate corners where its limited perception prevents it from identifying traversable paths. In contrast, Fig. 6(b) shows that DreamFlow achieves collision-free navigation by anticipating corridor layouts beyond its immediate sensor range, smoothly reaching the goal point (yellow marker) through proactive path adjustments.

The cluttered environment further highlights these performance differences. As shown in Fig. 6(c), the baseline collides with the wall segment that occludes the target position, unable to perceive a viable path around this obstacle until impact. Meanwhile, Fig. 6(d) demonstrates DreamFlow's successful collision-free navigation, where the extended terrain prediction enables the robot to plan an efficient path around the occluding wall from the beginning.

These real-world results confirm that the ability of DreamFlow to predict extended terrain from partial observations translates effectively to physical deployments, where real-world uncertainties such as sensor noise and more severe partial observability exist. While the baseline's reactive approach led to multiple collisions, our method maintained safe, collision-free trajectories throughout all trials, validating the practical value of predictive terrain modeling for real-world robot navigation.

VI. Conclusion

In this paper, we presented DreamFlow, an DRL-based framework that integrates CFM to enhance local navigation in a limited observation range. Our experiments demonstrated that CFM-based probabilistic latent prediction, which extrapolates from local observations to infer the extended environmental structure, effectively reduced redundant exploration, alleviated failures from local minima, and decreased collisions in cluttered environments. For future work, we plan to investigate the relationship between prediction horizon and navigation performance, integrate the training pipeline into an end-to-end framework, and explore multiple latent predictions for complex dynamic environments.

REFERENCES

[1] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, "Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 610–617, 2019.

[2] X. Kan, H. Teng, and K. Karydis, "Online exploration and coverage planning in unknown obstacle-cluttered environments," *IEEE Robot. Automat. Lett.*, vol. 5, no. 4, pp. 5969–5976, 2020.

[3] H. Kim, H. Kim, S. Lee, and H. Lee, "Autonomous exploration in a cluttered environment for a mobile robot with 2D-map segmentation and object detection," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 6343–6350, 2022.

[4] Z. Jian, S. Zhang, S. Chen, Z. Nan, and N. Zheng, "A global-local coupling two-stage path planning method for mobile robots," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 5349–5356, 2021.

[5] K. Chu, M. Lee, and M. Sunwoo, "Local path planning for off-road autonomous driving with avoidance of static obstacles," *IEEE Intell. Transp. Syst. Mag.*, vol. 13, no. 4, pp. 1599–1616, 2012.

[6] D. Lee, I. M. A. Nahrendra, M. Oh, B. Yu, and H. Myung, "TRG-planner: Traversal risk graph-based path planning in unstructured environments for safe and efficient navigation," *IEEE Robot. Automat. Lett.*, vol. 10, no. 2, pp. 1736–1743, 2025.

[7] F. Yang, C. Cao, H. Zhu, J. Oh, and J. Zhang, "FAR-Planner: Fast, attemptable route planner using dynamic visibility update," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2022, pp. 9–16.

[8] J. Liu, X. Chen, J. Xiao, S. Lin, Z. Zheng, and H. Lu, "Hybrid map-based path planning for robot navigation in unstructured environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2023, pp. 2216–2223.

[9] K. Nakhleh, M. Raza, M. Tang, M. Andrews, R. Boney, I. Hadžić, J. Lee, A. Mohajeri, and K. Palyutina, "SACPlanner: Real-world collision avoidance with a soft actor critic local planner and polar state representations," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 9464–9470.

[10] J. Lu, B. Tian, H. Shen, X. Zhang, and Y. Hui, "LPNet: A reaction-based local planner for autonomous collision avoidance using imitation learning," *IEEE Robot. Automat. Lett.*, vol. 8, no. 11, pp. 7058–7065, 2023.

[11] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter, "Advanced skills by learning locomotion and local navigation end-to-end," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2022, pp. 2497–2503.

[12] D. C. Conner, A. A. Rizzi, and H. Choset, "Composition of local potential functions for global robot control and navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, vol. 4, 2003, pp. 3546–3551.

[13] G. Klančar and M. Seder, "Combined stochastic-deterministic predictive control using local-minima free navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2021, pp. 5788–5793.

[14] T. Fuke, M. Endo, K. Honda, and G. Ishigami, "Towards local minima-free robotic navigation: Model predictive path integral control via repulsive potential augmentation," in *Proc. IEEE Int. Symp. Syst. Integr.*, 2025, pp. 1112–1117.

[15] J. Kim, S. Park, W. Lee, W. Kim, H. Choi, N. Doh, and C. Nam, "Escaping local minima: Hybrid artificial potential field with wall-follower for decentralized multi-robot navigation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2025, pp. 6616–6622.

[16] I. Meijer, M. Pantic, H. Oleynikova, and R. Siegwart, "Pushing the limits of reactive navigation: learning to escape local minima," *IEEE Robot. Automat. Lett.*, 2025.

[17] R. Guldenring, M. Görner, N. Hendrich, N. J. Jacobsen, and J. Zhang, "Learning local planners for human-aware navigation in indoor environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2020, pp. 6053–6060.

[18] U. Patel, N. K. S. Kumar, a. Sathyamoorthy, Adarsh Jahigh perform, and D. Manocha, "DWA-RL: Dynamically feasible deep reinforcement learning policy for robot navigation among mobile obstacles," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 6057–6063.

[19] L. Liu, D. Dugas, G. Cesari, R. Siegwart, and R. Dubé, "Robot navigation in crowded environments using deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2020, pp. 5671–5677.

[20] D. Hoeller, L. Wellhausen, F. Farshidian, and M. Hutter, "Learning a state representation and navigation in cluttered and dynamic environments," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 5081–5088, 2021.

[21] C. Zhang, J. Jin, J. Frey, N. Rudin, M. Mattamala, C. Cadena, and M. Hutter, "Resilient legged local navigation: Learning to traverse with compromised perception end-to-end," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2024, pp. 34–41.

[22] I. M. A. Nahrendra, B. Yu, M. Oh, D. Lee, S. Lee, H. Lee, H. Lim, and H. Myung, "Dreamwaq++: Obstacle-aware quadrupedal locomotion with resilient multimodal reinforcement learning," *IEEE Trans. Robot.*, vol. 42, pp. 819–836, 2026.

[23] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," in *Proc. Int. Conf. Learn. Represent.*, 2023.

[24] X. Liu, C. Gong, and Q. Liu, "Flow straight and fast: Learning to generate and transfer data with rectified flow," in *Proc. Int. Conf. Learn. Represent.*, 2023.

[25] Q. Dao, H. Phung, B. Nguyen, and A. Tran, "Flow matching in latent space," *arXiv preprint arXiv:2307.08698*, 2023.

[26] A. Tong, K. Fatras, N. Malkin, G. Huguet, Y. Zhang, J. Rector-Brooks, G. Wolf, and Y. Bengio, "Improving and generalizing flow-based generative models with minibatch optimal transport," *Trans. Mach. Learn. Res.*, 2024.

[27] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Automat. Mag.*, vol. 4, no. 1, pp. 23–33, 2002.

[28] Y. Jang, J. Baek, and S. Han, "Hindsight intermediate targets for mapless navigation with deep reinforcement learning," *IEEE Trans. Ind. Electron.*, vol. 69, no. 11, pp. 11 816–11 825, 2021.

[29] A. Debnath, G. J. Stein, and J. Košecká, "A hybrid approach to indoor social navigation: Integrating reactive local planning and proactive global planning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2025, pp. 10 432–10 438.

[30] M. S. Albergo and E. Vanden-Eijnden, "Building normalizing flows with stochastic interpolants," in *Proc. Int. Conf. Learn. Represent.*, 2023.

[31] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 6840–6851, 2020.

[32] K. Nguyen, A. T. Le, T. Pham, M. Huber, J. Peters, and M. N. Vu, "FlowMP: Learning motion fields for robot planning with conditional flow matching," *arXiv preprint arXiv:2503.06135*, 2025.

[33] Q. Zhang, Z. Liu, H. Fan, G. Liu, B. Zeng, and S. Liu, "FlowPolicy: Enabling fast and robust 3D flow-based policy via consistency flow matching for robot manipulation," in *Proc. Conf. Adv. Artif. Intell.*, vol. 39, no. 14, 2025, pp. 14 754–14 762.

[34] X. Zhai and C. Hao, "VFP: Variational flow-matching policy for multi-modal robot manipulation," *arXiv preprint arXiv:2508.01622*, 2025.

[35] S. Gode, A. Nayak, D. N. Oliveira, M. Krawez, C. Schmid, and W. Burgard, "FlowNav: Combining flow matching and depth priors for efficient navigation," *arXiv preprint arXiv:2411.09524*, 2024.

[36] L. Pinto, M. Andrychowicz, P. Welinder, W. Zaremba, and P. Abbeel, "Asymmetric actor critic for image-based robot learning," in *Robot. Sci. Syst.*, 2018.

[37] I. M. A. Nahrendra, B. Yu, and H. Myung, "DreamWaQ: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 5078–5084.

[38] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[39] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac Gym: High performance GPU based physics simulation for robot learning," in *Adv. Neural Inf. Process. Syst. (Datasets Benchmarks Track)*, 2021.

[40] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, *et al.*, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018.

[41] W. Xu and F. Zhang, "FAST-LIO: A fast, robust LiDAR-inertial odometry package by tightly-coupled iterated kalman filter," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 3317–3324, 2021.

[42] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic terrain mapping for mobile robots with uncertain localization," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 3019–3026, 2018.