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

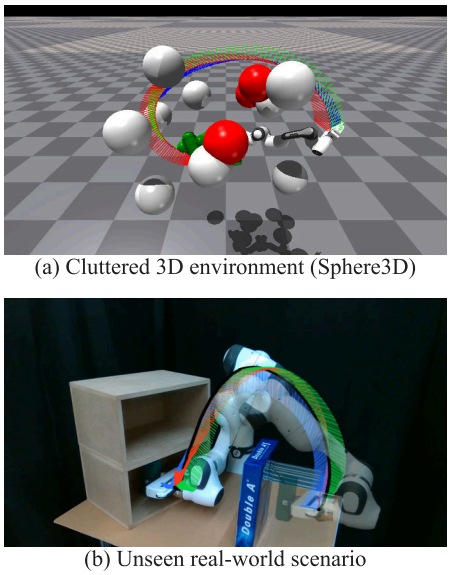
Prior-Constrained Explorative Guidance for Generalization in Diffusion Motion Planning

Sunhwi Kim[1], Junsu Kim[2], Seungjae Baek[2], Jaechan Shin[1], Jungeun Lee[1], Seongjae Lee[1], Kyungdon Joo[2†], Jeong hwan Jeon[1,2†]

***Abstract*— Diffusion-based planners have achieved generalization comparable to classical planners by leveraging inference-time optimization through guidance. However, their limited ability to capture environmental variations often constrains their responsiveness in unseen settings. In addition, the diversity-consistency trade-off inherent in guidance has remained unresolved. In this work, we propose Prior-Constrained Explorative Guidance (PCEG), a novel approach that gathers environmental information through local exploration and prevents guided samples from converging prematurely to similar solutions by leveraging a trajectory prior. The collected information is included in the guidance via stochastic gradient estimation, while a succinct parameter scheduling strategy enables latent optimization driven by environmental signals without significant computational overhead. Furthermore, during the modal-seeking stages of the reverse diffusion process, we employ a Gaussian Process (GP) to enforce dynamics-informed priors, effectively constraining the exploration region of each sample and thereby enhancing solution diversity. Across diverse benchmarks including 7-degree-of-freedom (7-DoF) robot-arm manipulation, PCEG substantially improves success rate by up to 30 percentage points compared to competitive diffusion planners without compromising trajectory quality, even in scenarios involving unseen obstacles. Real-world experiments further validate these findings, showcasing the generation of smooth, collision-free trajectories in novel environments. The project page is available at https://rml-unist.github.io/PCEG/.**

I. INTRODUCTION

Motion planning constitutes a fundamental research area in robotics [1]–[3]. The primary goal of motion planning is to compute trajectories that connect initial configurations to desired goals while satisfying feasibility constraints (*e.g.*, collision avoidance) and optimality criteria (*e.g.*, smoothness). This field has been approached through two broad families of methods: classical planners and learning-based planners. Classical planners [4]–[6] offer solutions in various environments and a broad generalization without requiring training. However, they can be computationally prohibitive in cluttered scenarios or high-dimensional spaces and may yield dynamically infeasible trajectories, complicating real-world deployment. Meanwhile, learning-based methods [7]–[9] produce safer and smoother trajectories by leveraging learned distributions from demonstrations or classical solutions. Nevertheless, their performance is inherently bounded by training data, potentially limiting generalization to novel environments.



(a) Cluttered 3D environment (Sphere3D)

(b) Unseen real-world scenario

Fig. 1: Our guidance method generates safe and smooth trajectories with additional obstacles colored in red (a), and unencountered real-world environment (b). This generalization does not require extra training with our proposed guidance method. Local exploration is added to the naive classifier guidance and modal search is diversified through GP noise blended during modal-seeking stages of the reverse diffusion process.

Recently, diffusion planners [10], [11] have emerged as a promising solution to alleviating this trade-off, improving generalization through multimodal trajectory modeling and guidance [12], [13]. The guidance steers the generative process toward satisfying criteria and safety constraints. Conditioned on guidance, diffusion planners can generate diverse trajectories when the environment admits multiple feasible paths. This capability expands user choice and improves adaptability to environmental change and modeling. However, guidance typically entails a trade-off between sample diversity and consistency with the conditioning signals [12]. This suggests considerable room for improving diffusion planners, particularly through advances in classifier guidance.

This work was supported by the InnoCORE program of the Ministry of Science and ICT (#N10250155), by the National Research Foundation of Korea (NRF) grants funded by the Korean government (MSIT) (No.RS-2022-NR070854 and No.RS-2025-02216916), and by the Institute of Information & communications Technology Planning & Evaluation (IITP) grants funded by the Korean government (MSIT) (No.RS-2020-II201336, Artificial Intelligence graduate school support (UNIST); and No.RS-2024-00341055).

[1] Department of Electrical Engineering, Ulsan National Institute of Science and Technology (UNIST), Ulsan, South Korea.

[2] Artificial Intelligence Graduate School (AIGS), Ulsan National Institute of Science and Technology (UNIST), Ulsan, South Korea.

† Co-corresponding authors: {kyungdon, jhjeon}@unist.ac.kr

In this work, we aim to improve the guidance that simultaneously enhances generalization and trajectory diversity. We propose Prior-Constrained Explorative Guidance (PCEG), which gathers information about environmental changes via local exploration and constrains each sample to informative regions of the search space. Concretely, to promote generalization, we sample perturbations in the neighborhood of a candidate trajectory, evaluate their costs, and estimate stochastic gradients [14]. To avoid intensive computation in universal guidance [15], we derive a simple yet effective parameter schedule. To increase diversity, we blend GP noise [5] with Gaussian noise, thereby constraining each sample to dynamics-informative regions during modal-seeking stages. Across diverse environments, the proposed method yields a substantial increase in success rate without degrading path quality. Moreover, it enables smooth and safe real-world deployment in an unseen environment, as presented in Fig. 1 (b). The main contributions are summarized below:

- We propose a guidance mechanism that generates diverse trajectories in out-of-distribution environments by local exploration and GP prior-constrained sampling.
- Our method incorporates information on environmental changes via stochastic gradient estimation and efficient parameter scheduling, then blends the GP prior into the pretrained diffusion process to restrict search space during modal-seeking stages.
- We evaluate our method across diverse simulated environments and an unseen real-world setting where other diffusion planners have difficulty handling, demonstrating high success rates and solution diversity without sacrificing path quality, even with a computationally light variant.

II. Related Works

A. Classical planners

Sampling-based motion planning methods [16]–[18], such as Rapidly-exploring Random Trees (RRT) [19] and RRT-Connect (RRTC) [4], efficiently explore the search space through random sampling. These methods are theoretically supported by probabilistic completeness, which ensures that a solution trajectory will be found if one exists. However, the computational complexity of these planners tends to increase significantly with higher-dimensional state spaces, and additional trajectory post-processing is typically required to enhance smoothness.

Trajectory optimization methods [20], [21], including Gaussian Process Motion Planning (GPMP) [5], explicitly incorporate dynamic feasibility constraints into their optimization objectives. Nonetheless, these methods can get stuck in poor local minima and may produce trajectories in collision. Two common remedies for these issues are stochastic gradient estimation, exemplified by Stochastic Trajectory Optimization for Motion Planning (STOMP) [14], and optimization over a diverse set of initial guesses. While these classical planners exhibit adaptability to novel environments, they can still face significant challenges in highly cluttered spaces, and introducing diversity may inadvertently increase the likelihood of producing dynamically infeasible trajectories.

B. Learning-based planners fused with classical methods

Learning-based planners are capable of generating smooth trajectories even in complex environments [9]; however, handling distributional shifts remains challenging [22]. Two common strategies to mitigate this limitation are warm-start methods and diffusion-based planners. Warm starts [6], [7], [23] initialize trajectories from learned priors, subsequently refining them through optimization. Diffusion planners [10], [11], [24] integrate sampling and optimization through guidance. Classical optimization principles are frequently incorporated to improve guidance: Diffusion-ES [25] employs evolutionary strategies on multiple samples, enhancing global exploration, while EDMP [26] guides unconditional trajectory distributions using an ensemble of cost functions that capture diverse environmental perspectives.

In our method, local exploration acquires additional environmental information, and each sample searches for trajectory modals within a dynamically constrained latent space, which is independent of exploring different viewpoints of the same environment.

III. Method

In this section, we propose *Prior-Constrained Explorative Guidance (PCEG)*, a novel motion planning framework based on diffusion models, designed to generate collision-free and dynamically feasible trajectories in diverse environments. In Sec. III-A, we first define the problem and provide an overview of the proposed method. We also present the background on diffusion with classifier guidance that accounts for obstacles and trajectory smoothness. We then discuss the two core components of our method: explorative guidance and prior-constrained sampling in Sec. III-B and III-C, respectively.

A. Overview

We aim to generate a trajectory $\boldsymbol{\tau} = [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_H]$, a sequence of joint configurations $\mathbf{q} \in \mathbb{R}^N$, that avoids obstacles $\mathcal{O}$, given start, $\mathbf{q}_1$, and goal, $\mathbf{q}_H$, configurations. Here, H is the horizon of a trajectory and N is the dimension of a robot configuration space. To simultaneously enhance both generalization performance and trajectory diversity, we propose a novel motion planning method that collects environment information through local exploration while leveraging trajectory priors to prevent guided samples from prematurely converging to similar solutions.

Classifier-guided diffusion. In this work, we model the motion planner using a classifier-guided diffusion model that enhances the generalization of diffusion planners to unseen environments [10], [11], [26]. Basically, the forward process with T_{diff} diffusion timesteps adds Gaussian noise, $\mathcal{N}(\mathbf{0}, \mathbf{I})$, into a clean trajectory $\boldsymbol{\tau}^0$:

$$\boldsymbol{\tau}^i = \sqrt{\bar{\alpha}^i}\boldsymbol{\tau}^0 + \epsilon\sqrt{1-\bar{\alpha}^i},\ \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{1}$$

where $\bar{\alpha}^i = \prod_{j=0}^{i-1} \alpha^j$, $\{\alpha^j\}_{j=0}^{T_{\text{diff}}-1}$ is pre-scheduled. The diffusion model learns the reverse process by predicting the added noise via the standard denoising objective:

$$\mathcal{L}_\theta(\epsilon, \epsilon_\theta) = \mathbb{E}_{i\sim[0,T_{\text{diff}}-1],\ \boldsymbol{\tau}^i\sim q(\boldsymbol{\tau}^i|\boldsymbol{\tau}^0)} \left[\left\| \epsilon - \epsilon_\theta(\boldsymbol{\tau}^i) \right\|^2 \right], \quad (2)$$

where $q(\boldsymbol{\tau}^i|\boldsymbol{\tau}^0)$ is the forward process and ϵ denotes the same Gaussian noise used in $q(\boldsymbol{\tau}^i|\boldsymbol{\tau}^0)$. Diffusion planners optimize trajectories by guidance at inference time.

The classifier guidance approximates a conditioned posterior mean with a sum of prior and likelihood scores for enhancing the generalization performance:

$$\hat{\epsilon}_\theta = \epsilon_\theta(\boldsymbol{\tau}^i) - s\sqrt{1-\bar{\alpha}^i}\nabla_{\boldsymbol{\tau}^i} \log y_\phi(\boldsymbol{\tau}^i, \mathbf{e}), \quad (3)$$

where s is a guidance scale and $y_\phi(\boldsymbol{\tau}^i, \mathbf{e})$ quantifies how well a noisy trajectory $\boldsymbol{\tau}^i$ satisfies conditions $\mathbf{e}$. To avoid training a noisy-input network, [15] and [27] use an estimated clean sample, $\hat{\boldsymbol{\tau}}^0$:

$$\hat{\boldsymbol{\tau}}^0 = \frac{\boldsymbol{\tau}^i - \sqrt{1-\bar{\alpha}^i}\epsilon_\theta(\boldsymbol{\tau}^i)}{\sqrt{\bar{\alpha}^i}}. \quad (4)$$

This simple modification broadens the applicability of diffusion models to flexible conditions. Additionally, [15] introduces backward guidance:

$$\tilde{\epsilon}_\theta = \hat{\epsilon}_\theta - s\sqrt{\frac{\bar{\alpha}^i}{1-\bar{\alpha}^i}}\Delta^*, \quad \Delta^* = \arg\min_{\Delta}\ E(\hat{\boldsymbol{\tau}}^0 + \Delta, \mathbf{e}), \quad (5)$$

where $E(\cdot,\cdot)$ can be any energy function related to $y_\phi(\cdot,\cdot)$. These additional steps strengthen the consistency with conditions in the diffusion process, but require backpropagation through the model and the steepest-descent direction.

Collision and smoothness. To obtain collision-free and dynamically feasible trajectories through guidance, both smoothness and collision are expressed via differentiable cost functions, following [11].

Let the obstacle set $\mathcal{O}$ be defined as a union of infeasible configurations in an environment. Then, collision with obstacles is expressed with the signed distance field for predefined collision spheres $\mathbf{S} = \{\mathbf{S}_k\}_{k=1}^K$ from a subset of robot joints:

$$\begin{aligned} c_{obs}(\mathbf{q}) &= \sum_{k=1}^{K} c_{sdf}(\mathcal{F}(\mathbf{q}, \mathbf{S}_k), \mathcal{O}), \\ c_{sdf}(\mathcal{F}(\mathbf{q}, \mathbf{S}_k), \mathcal{O}) &= \max(-d(\mathcal{F}(\mathbf{q}, \mathbf{S}_k), \mathcal{O}) + \delta, 0), \end{aligned} \quad (6)$$

where K is the cardinality of $\mathbf{S}$, $\mathcal{F}(\cdot,\cdot)$ is a differentiable forward kinematics, $d(\cdot,\cdot)$ is the signed distance field, and δ is a margin. For each $\mathbf{q}$, the collision cost is added over all collision spheres, as in Eq. (6).

For a manipulator, self-collision is also considered, by pairing each $\mathbf{S}_k$ with L other spheres, $\{\mathbf{S}_{k,l}\}_{l=1}^L \subset \mathbf{S}/\{\mathbf{S}_k\}$. Collision cost for each pair, $(\mathbf{S}_k, \mathbf{S}_{k,l})$, is calculated with an L2 distance with margin δ. Self-collision costs for all pairs are summed for the final cost:

$$\begin{aligned} &c_{self}(\mathbf{q}) = \\ &\sum_{k=1}^{K}\sum_{l=1}^{L} \max(-\left\|\mathcal{F}(\mathbf{q}, \mathbf{S}_k) - \mathcal{F}(\mathbf{q}, \mathbf{S}_{k,l})\right\|_2 + \delta, 0). \end{aligned} \quad (7)$$

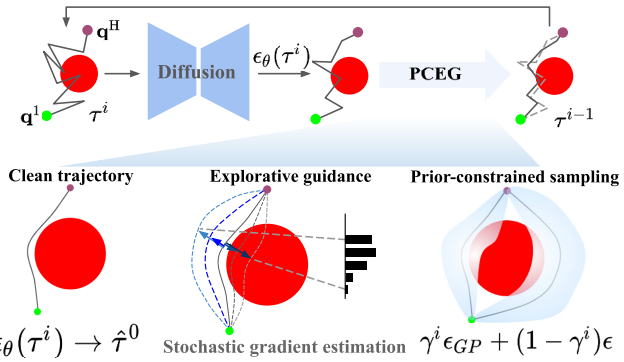


Fig. 2: An overview of our proposed method (the red circle denotes an obstacle). After a diffusion model denoises trajectories, PCEG guides each sample. Based on estimated clean trajectories, explorative guidance enhances trajectories via stochastic gradient estimation. Then, a GP prior constrains reverse diffusion sampling.

Smoothness is formulated by GP cost [5]:

$$c_{GP}(\mathbf{x}_t, \mathbf{x}_{t+1}) = \frac{1}{2}\sum_{t=1}^{H-1} \left\|\Phi_{t,t+1}\mathbf{x}_t - \mathbf{x}_{t+1}\right\|_{\mathbf{Q}_{t,t+1}}, \quad (8)$$

where $\mathbf{x}_t = [\mathbf{q}_t^T, \dot{\mathbf{q}}_t^T]^T$, $\|\cdot\|_{\mathbf{Q}_{t,t+1}}$ is a Mahalanobis distance,

$$\Phi_{t,t+1} = \begin{bmatrix} \mathbf{I} & \Delta t\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}, \quad \mathbf{Q}_{t,t+1} = \begin{bmatrix} \frac{1}{3}\Delta t^3\mathbf{Q}_c & \frac{1}{2}\Delta t^2\mathbf{Q}_c \\ \frac{1}{2}\Delta t^2\mathbf{Q}_c & \Delta t\mathbf{Q}_c \end{bmatrix},$$

Δt is a timestep, and $\mathbf{Q}_c$ is a power spectral density. The total cost function $\mathbf{c}$ is the weighted sum of these costs. To illustrate, for 7-DoF manipulation,

$$\mathbf{c} = w_{obs}c_{obs} + w_{self}c_{self} + w_{GP}c_{GP}, \quad (9)$$

where w_{obs}, w_{self}, and w_{GP} balance each cost term.

Although the classifier guidance makes valid trajectories with $\mathbf{c}$, this naive method leads to insufficient information on environmental variations and leaves the unresolved trade-off between diversity and consistency. To handle these problems, our method makes guidance explorative and the search space constrained with dynamics-aware noise, as depicted in Fig. 2.

Our method predicts a clean trajectory, $\hat{\boldsymbol{\tau}}^0$, using the noise predicted by the diffusion model. We then sample perturbations around $\hat{\boldsymbol{\tau}}^0$ and estimate a cost-based gradient from the perturbed trajectories. After a guidance step, we blend GP noise with Gaussian noise, which constrains dispersion from the guided mean using the structured covariance of GP. In this way, our method generates diverse trajectories in unencountered environments.

B. Explorative guidance (EG)

To enhance generalization via collection of environmental information, our method extends the classifier guidance by incorporating stochastic gradient estimation and backward universal guidance.

Stochastic gradient estimation. The stochastic gradient estimation [14] samples noisy trajectories and weights them

by softmax cost:

$$\mathbb{E}\left[\nabla_{\boldsymbol{\tau}}\mathbf{c}(\boldsymbol{\tau}+\delta\tau)\right] \approx -\sum_{n=1}^{N_p} \text{softmax}\left(-\frac{1}{\lambda}\mathbf{c}\left(\boldsymbol{\tau}+\delta\tau^n\right)\right)\delta\tau^n, \tag{10}$$

where N_p is the number of perturbations, λ is the temperature, and $\delta\tau$ is a sampled perturbation.

In our method, explorative gradient is combined with the classifier guidance via a parameter scheduling grounded on the backward universal guidance, under the assumption of the detachment of predicted noise:

$$\begin{aligned}\hat{\epsilon}_\theta &= \epsilon_\theta - s\left(\sqrt{1-\bar{\alpha}^i}+\frac{1}{\sqrt{1-\bar{\alpha}^i}}\right)\sum_{n=1}^{N_p} w^n\delta\tau^n,\\ w^n &= \text{softmax}\left(-\frac{1}{\lambda}\mathbf{c}(\hat{\boldsymbol{\tau}}^0+\delta\tau^n)\right).\end{aligned} \tag{11}$$

Derivation. When guidance and trajectory optimization start from the same latent vector, the forward and backward guidance can be combined in a single equation, using Eq. (4) and (5):

$$\hat{\epsilon}_\theta = \epsilon_\theta + s\sqrt{1-\bar{\alpha}^i}\nabla_{\boldsymbol{\tau}^i}\mathbf{c}(\hat{\boldsymbol{\tau}}^0) + s\sqrt{\frac{\bar{\alpha}^i}{1-\bar{\alpha}^i}}\nabla_{\hat{\boldsymbol{\tau}}^0}\mathbf{c}(\hat{\boldsymbol{\tau}}^0), \tag{12}$$

where the likelihood of the condition is designed as $\exp\left(-\mathbf{c}(\hat{\boldsymbol{\tau}}^0)\right)$, and the one-step gradient descent, $-\nabla_{\hat{\boldsymbol{\tau}}^0}\mathbf{c}\left(\hat{\boldsymbol{\tau}}^0\right)$, replaces the steepest direction. With the stochastic gradient estimation,

$$\nabla_{\boldsymbol{\tau}^i}\mathbf{c}(\hat{\boldsymbol{\tau}}^0) \approx -\sum_{n=1}^{N_p} w^n\delta\tau^{i,n},\ \nabla_{\hat{\boldsymbol{\tau}}^0}\mathbf{c}(\hat{\boldsymbol{\tau}}^0) \approx -\sum_{n=1}^{N_p} w^n\delta\tau^{0,n}. \tag{13}$$

A constant multiple relates the perturbations in the two spaces, if the predicted noise is detached:

$$\hat{\boldsymbol{\tau}}^0 + \delta\tau^0 = \frac{\boldsymbol{\tau}^i + \delta\tau^i - \sqrt{1-\bar{\alpha}^i}\epsilon_\theta}{\sqrt{\bar{\alpha}^i}}, \quad \delta\tau^0 = \frac{1}{\sqrt{\bar{\alpha}^i}}\delta\tau^i. \tag{14}$$

After Eq. (13) and (14) are substituted into Eq. (12), Eq. (11) is derived.

C. Prior-constrained (PC) sampling

To resolve the innate trade-off in classifier guidance, sampling in the reverse diffusion process is constrained by dynamics-informative prior during modal-seeking stages, improving trajectory variance.

GP prior. GP is a collection of random variables whose samples are in a joint Gaussian distribution, $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\mathcal{K}})$. The covariance $\boldsymbol{\mathcal{K}}$ can be structured based on a linear time-varying stochastic differential equation:

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{u}(t) + \mathbf{F}(t)\mathbf{w}(t), \tag{15}$$

where $\mathbf{x}(t)$ is a system state, $\mathbf{A}(t)$ and $\mathbf{F}(t)$ are time-varying systems, $\mathbf{u}(t)$ is a control input, $\mathbf{w}(t)$ is a GP white noise, $\mathbf{w}(t) \sim \mathcal{GP}(\mathbf{0}, \mathbf{Q}_c\delta(t-t'))$, and $\delta(\cdot)$ is a Dirac delta function. With a constant-velocity model, the same $\Phi_{t,t+1}$ is obtained with Eq. (8). A GP prior with the structured $\boldsymbol{\mathcal{K}}$ gives dynamics-informative smooth samples [5], [6].

Algorithm 1 Prior-Constrained Explorative Guidance (PCEG) : Expandable to batch operations

Input: Pretrained diffusion model θ, GP $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\mathcal{K}})$, start and goal configurations $(\mathbf{q}_1, \mathbf{q}_H)$, total cost function $\mathbf{c}$, guidance scale s, scheduling terms $(\bar{\alpha}^i, \sigma^i)$
$\boldsymbol{\tau}^{T_{\text{diff}}} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $i = T_{\text{diff}}-1, \ldots, 0$ **do**
  ▷ *Predict* $\hat{\mathbf{x}}^0$ *from the latent,* $\mathbf{x}^i$*: Eq.* (4)
  $\hat{\boldsymbol{\tau}}^0 = \frac{\boldsymbol{\tau}^i - \sqrt{1-\bar{\alpha}^i}\epsilon_\theta(\boldsymbol{\tau}^i)}{\sqrt{\bar{\alpha}^i}}$
  ▷ *Exploratory guidance: Eq.* (11)
  $\hat{\epsilon}_\theta = \epsilon_\theta - s\left(\sqrt{1-\bar{\alpha}^i}+\frac{1}{\sqrt{1-\bar{\alpha}^i}}\right)\sum_{n=1}^{N_p} w^n\delta\tau^n$
  ▷ *Blend GP noise with cosine annealing: Eq.* (16)
  $\epsilon_{\text{GP}} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\mathcal{K}}),\ \gamma^i = 1 - \cos(\frac{i}{T_{\text{diff}}}\frac{\pi}{2})$
  $\epsilon^i_{\text{blend}} = \gamma^i\epsilon_{\text{GP}} + (1-\gamma^i)\epsilon$
  ▷ *Perturb the guided mean with the blended noise*
  $\boldsymbol{\tau}^i := \mu^i(\hat{\epsilon}_\theta) + \sigma^i\epsilon^i_{\text{blend}}$
  ▷ *Re-apply start and goal configurations*
  $\boldsymbol{\tau}^i_1 = \mathbf{q}_1,\ \boldsymbol{\tau}^i_H = \mathbf{q}_H$
end for
Output: Optimized trajectory $\boldsymbol{\tau}$

Stages in a guided diffusion. Guided diffusion can be divided into three stages with respect to the influence of guidance [27]. In the initial chaotic stage, updates are large, but the guidance barely makes anything reasonable. During the semantic stage, guidance produces substantial, content-level changes. In the final refinement stages, guidance adjusts only details of the result. Accordingly, modal-seeking behavior occurs in the chaotic and semantic stages.

Our method blends the GP noise, ϵ_{GP}, with the Gaussian noise during the modal-seeking stages. It constrains exploration to a region of smooth trajectories, mitigating diversity reduction induced by guidance. As sampling approaches the state space, the generated trajectories remain diverse, driven by the placement of obstacles. To keep the semantic stage close to the trained reverse dynamics for controllability, cosine scheduling is adopted on the blending weight:

$$\begin{aligned}\gamma^i &= 1 - \cos(\frac{i}{T_{\text{diff}}}\frac{\pi}{2}),\ i \in \{0, ..., T_{\text{diff}}-1\},\\ \epsilon^i_{\text{blend}} &= \gamma^i\epsilon_{\text{GP}} + (1-\gamma^i)\epsilon.\end{aligned} \tag{16}$$

Algorithm 1 presents the whole process of our method.

IV. Experiment

This section examines our method in four aspects: 1) performance gains over strong baselines across diverse environments. 2) effect of each component: the contribution of local exploration to robustness under environmental changes, the benefit of guidance-time optimization via parameter scheduling relative to the warm start strategy, and the effect of the PC sampling on diversity. 3) the impact of computationally efficient variants on performance. 4) generalization to novel real-world settings.

To investigate these aspects, Sec. IV-A describes the experimental setup and evaluation metrics, Sec. IV-B introduces competitive baselines, Sec. IV-C presents quantitative and qualitative comparisons, Sec. IV-D performs ablation studies of each component, Sec. IV-E introduces computation reduction techniques and assesses their impact, and Sec. IV-F reports real-world deployment results.

A. Experiment setting

We generate trajectories for a point mass ($N = 2$) in 2D and a Franka Panda ($N = 7$) in 3D environments with unseen obstacles (Fig. 3). For collision costs, the margin δ is set as 0.05 for a point mass, and as 0.06 for the manipulator, considering the link volume. There is no manipulator-specific cost (*e.g.* end-effector cost).

Metrics (averaged over 300 seeds) include planning time (T), success rate (S), path length (PL), path smoothness (PS), variance (V), and collision intensity (I). For S, success is declared if at least one collision-free trajectory is found. T is considered only for collision-free plans to avoid overestimation from failures. T is not compared because certain hyperparameters (*e.g.*, the number of whole diffusion loops) were tuned solely for S. With the definition,

$$\text{collision}(\mathbf{q}) = \begin{cases} 1, & \exists\ \mathbf{S}_k \in \mathbf{S},\ d(\mathcal{F}(\mathbf{q}, \mathbf{S}_k), \mathcal{O}) < 1\text{e}^{-2} \\ 0, & \text{otherwise}, \end{cases} \tag{17}$$

other metrics are defined as,

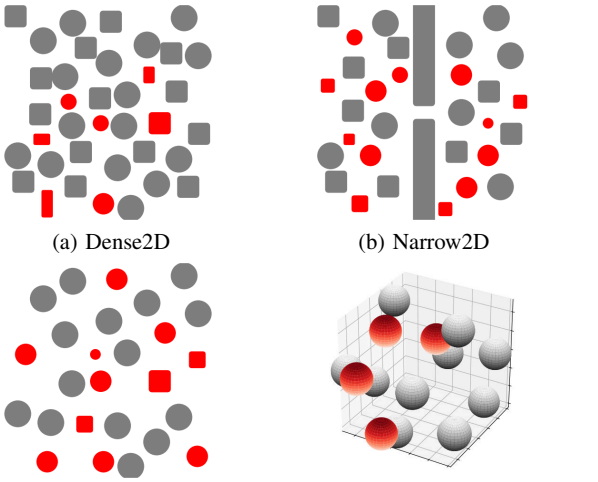
$$PL = \sum_{t=1}^{H-1} \|\mathbf{q}_t - \mathbf{q_{t+1}}\|_2\,,\ PS = \sum_{t=1}^{H} \|\dot{\mathbf{q}}_t\|_2\,,$$
$$I = \frac{1}{BH} \sum_{b=1}^{B} \sum_{t=1}^{H} \text{collision}(\mathbf{q}_{b,t}),\ V = \sum_{t=1}^{H} \text{Var}\left(\text{D}(\mathbf{q}_t)\right), \tag{18}$$

where $B = 100$, $D(\mathbf{q}_t) = \{||\mathbf{q}_t^k - \mathbf{q}_t^{k'}||_2 \mid 0 \leq k < k' \leq B_s - 1\}$, and B_s is the number of collision-free trajectories in a batch. All experiments are conducted on an AMD Ryzen 9 5950X and a single NVIDIA RTX 3090 GPU.

B. Baselines

Our method is compared against both classical and diffusion baselines. Classical methods include RRTC [4] and GPMP [5]. These two methods collect training data for pretrained MPD [11]. For diffusion planners, MPD and Diffusion-ES [25] are selected. Diffusion-ES resembles our method in that it searches samples via global exploration, but it enhances them by selecting elites rather than exploiting gradients. Hyperparameters are adjusted until each baseline obtains competitive success rate, regardless of planning time.

For PCEG, we use an off-the-shelf pretrained MPD model based on a U-Net [28] with noise parameterization. For the explorative guidance, perturbations are drawn from the GP prior and scaled by $\sqrt{1-\bar{\alpha}^i}$. We set $N_p = 5$ for 2D environments and $N_p = 15$ for the manipulation. To mimic partial gradients, per-configuration costs are not summed; their gradients are estimated independently. Start/goal configurations are enforced at each diffusion timestep. To prioritize smoothness, we clip the guidance scale at 0.5 during 5 additional noiseless sampling, slightly lowering the success rate but improving path quality.



(a) Dense2D (b) Narrow2D (c) Simple2D (d) Sphere3D w. Franka Panda

Fig. 3: Environments for experiments. Unseen obstacles are highlighted in red. The environments were selected to span diverse obstacle densities, permitted multimodality, width of cost plateaus and robots.

While other algorithms benefit from GPU-based parallelism and thus run quickly, RRTC is inherently sequential and therefore incurs longer runtimes. To address this, following [11], we utilized a GPU-accelerated implementation of RRTC. In particular, collision-free configurations are precomputed to exploit parallelized collision checking, and nearest-neighbor searches are also parallelized on the GPU. RRTC is evaluated using this optimized implementation.

Since RRTC returns a single path on success, V and I are not measured. RRTC uses 15 seconds, set via Simple2D calibration. GPMP runs 30 iterations to match planning time with MPD at most. For MPD and our method, we increase the guidance scale until path quality visibly deteriorates in a small seed set, then select a scale with the best success rate after conducting a sweep from 0.005 to 7. For Diffusion-ES, temperature is swept from 0.05 to 10 and the number of whole loops is set to 20 with competitive success rate.

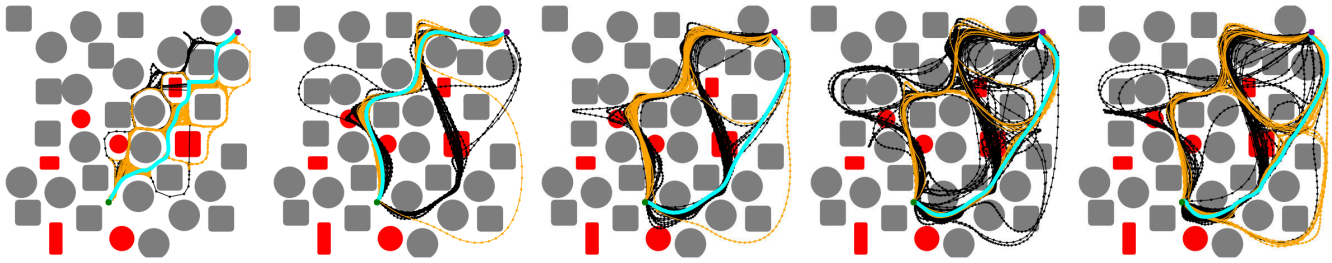
C. Performance comparison

Quantitative comparison. Our method is compared with selected baselines, detailed in Table I. Classical planners generally find feasible trajectories with sufficient computation, but are degraded under tight time budgets. Especially, RRTC fails to make collision-free trajectories under dense obstacles or high-dimensional state spaces, despite accelerated implementation. With a large batch size, GPMP presents strong generalization indicated by high S and low I but it does not improve trajectory diversity and dynamic feasibility. Diffusion planners give various smooth trajectories using their multi-modal prior, but they show lower S than GPMP in the presence of unseen obstacles even with their own

TABLE I: Comparisons between baselines and the proposed method, with the best highlighted in bold and the second-best underlined. Each baseline shows uneven performance on S and V, whereas our method attains strong results in both metrics, without sacrifice on path quality. ‡We increase batch size to match diffusion planners. Alleviating the initial-trajectory issue, GPMP attains high success rate but dynamic feasibility and trajectory diversity are not improved.

Environment	Dense2D						Narrow2D					
Method	T[s] ↓	S[%] ↑	PL ↓	PS ↓	V ↑	I[%] ↓	T[s] ↓	S[%] ↑	PL ↓	PS ↓	V ↑	I[%] ↓
RRTC	-	0.0	-	-	-	-	8.7669	0.33	**1.5201**	5.9056	-	-
GPMP‡	0.4043	**96.0**	**1.8378**	8.4064	0.8820	**1.0808**	0.4053	82.33	1.7603	4.7546	0.7428	**1.4458**
MPD	0.3724	81.33	1.9844	3.3171	1.6947	7.6422	0.3743	70.67	1.7061	**2.1086**	0.4003	9.0434
Diffusion-ES	2.8869	82.33	1.9746	3.3704	2.0088	9.3982	2.9012	72.0	1.7171	2.1432	0.5039	9.6705
PCEG (Ours)	0.5735	85.0	1.9526	**3.2559**	**3.4008**	8.3730	0.5733	**92.0**	1.7478	2.1584	**1.1874**	8.2395

Environment	Simple2D						Sphere3D w. Franka Panda					
RRTC	2.6660	97.67	2.9223	15.2386	-	-	3.6539	30.0	10.1967	58.8870	-	-
GPMP‡	0.4121	**100.0**	**1.5736**	3.9179	0.9617	**0.0671**	10.3610	81.67	6.8792	16.5661	6.1449	**0.1598**
MPD	0.3781	82.67	1.6015	**2.0526**	1.1343	13.9184	9.2180	59.33	**5.9304**	**5.5145**	11.1329	18.1603
Diffusion-ES	2.9090	84.33	1.6009	2.1274	1.6345	14.9705	4.3481	63.0	5.9504	5.5756	18.5080	18.6252
PCEG (Ours)	0.5743	98.67	1.5870	2.0952	**3.0401**	6.8019	10.3148	**93.0**	6.0188	5.5957	**23.9041**	6.4153



(a) GPMP (b) MPD (c) Diffusion-ES (d) EG (Ours) (e) PCEG (Ours)

Fig. 4: Qualitative results for baseline planners in Dense2D. Yellow lines are collision-free trajectories, but black lines are in collision. The cyan line indicates the best (PL + PS). EG indicates PCEG without prior-constrained sampling.

guidance. MPD exhibits low S in 7-DoF planning under harsh conditions, inflated collision costs and no informative end-effector signal. In addition, Diffusion-ES requires loops on full diffusion timesteps to obtain competitive success rate. Unlike other methods, PCEG shows high S and V, especially in Narrow2D where exploration is critical due to wide plateaus in SDF. Although GPMP sacrifices PS to increase S in Dense2D, our method finds diverse smooth trajectories, indicated in PS and V. Given that dense obstacles lead to highly curved avoidance trajectories, this result suggests that our method produces dynamically feasible trajectories by exploring diverse candidates and preventing premature convergence. Overall, our method yields the highest trajectory diversity while maintaining path quality and success rates comparable to baselines across all environments, showing improved generalization.

Qualitative comparison. Fig. 4 qualitatively compares the baselines. Trajectories planned by GPMP are highly jerky and unsafe, exploiting severely narrow gaps. However, all diffusion planners produce smooth trajectories between wider spaces. Although Diffusion-ES enables global exploration, elite sampling makes its search space similar to MPD, as indicated by the black lines. In contrast, PCEG achieves more diverse exploration, discovering more various collision-free trajectory modes. As indicated in the cyan trajectory, it is helpful to search diverse modals for obtaining a trajectory closer to the desired criteria.

D. Ablation study

This section examines each component of our method, EG and PC sampling. The result is illustrated in Table II. The ablation study is conducted on Dense2D with the densest obstacles and Sphere3D with the 7-DoF manipulator.

Explorative guidance. EG is compared with MPD and MPD warm start (MPDws) to examine the effectiveness of exploration in guidance and whether EG offers an advantage over post-optimization. In MPDws, MPD offers initial trajectories for stochastic post-optimization. EG consistently shows higher S and V than MPD with on par path quality, indicating that local exploration in guidance increases generalization in unseen obstacles, generating diverse trajectories.

Leveraging stochastic post-optimization, MPDws consistently outperforms MPD across S, V, and I. EG, in turn, incorporates the benefits of post-exploration into the guidance process via the parameter scheduling. EG exhibits weaker collision handling without fine local adjustment in state space, but instead produces a more diverse set of high-quality trajectories with comparable S, searching for trajectory modals in the latent space. Qualitatively, EG yields a more diverse exploration than MPD, as shown in Fig. 4d.

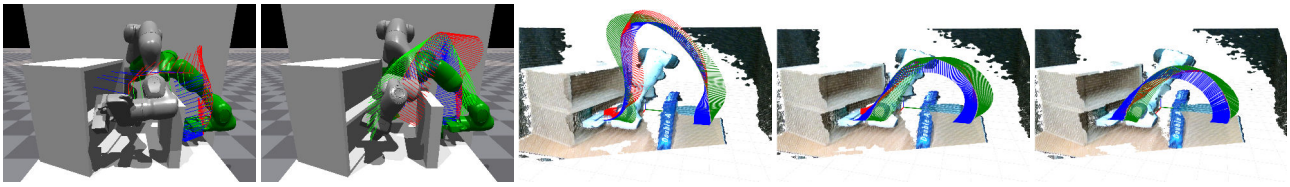
Prior-constrained sampling. Table II also presents performance difference between PCEG and EG, Δ(vs. EG). In spite of the modest performance degradation, diversity

TABLE II: Ablation study on explorative guidance and prior-constrained sampling. *Comparison of stochasticity in post-optimization and guidance. 15 additional iterations are executed after MPD. † Performance gap between PCEG and EG. Differences in S and I denote percentage points. For S and V, [+] denotes improvement over EG, [-] for the others.

Environment	Dense2D						Sphere3D w. Franka Panda					
Method	T[s] ↓	S[%] ↑	PL ↓	PS ↓	V ↑	I[%] ↓	T[s] ↓	S[%] ↑	PL ↓	PS ↓	V ↑	I[%] ↓
MPD	0.3724	81.33	1.9844	3.3171	1.6947	7.6422	9.2180	59.33	**5.9304**	**5.5145**	11.1329	18.1603
MPDws-15*	0.5499	**85.67**	1.9970	3.6253	2.3015	**2.7015**	14.7033	81.33	6.5712	6.9145	13.4514	**2.5909**
EG (Ours)	0.5697	**85.67**	**1.9302**	**3.2343**	3.0303	8.8638	10.3162	88.33	6.0124	5.5420	15.5410	6.1188
PCEG (Ours)	0.5735	85.0	1.9526	3.2559	**3.4008**	8.3730	10.3148	**93.0**	6.0188	5.5957	**23.9041**	6.4153
Δ(vs. EG)†	-	-0.67	+0.0224	+0.0216	+0.3705	-0.4908	-	+4.67	+0.0064	+0.0537	+8.3631	+0.2965

TABLE III: Results after computation reduction. The comparison is conducted among our PCEG variants.

Environment	Dense2D						Sphere3D w. Franka Panda					
Method	T[s] ↓	S[%] ↑	PL ↓	PS ↓	V ↑	I[%] ↓	T[s] ↓	S[%] ↑	PL ↓	PS ↓	V ↑	I[%] ↓
MPD	0.3724	81.33	1.9844	3.3171	1.6947	7.6422	9.2180	59.33	5.9304	5.5145	11.1329	18.1603
MPD-DDIM	0.1965	79.0	2.0091	3.5266	2.2145	12.5048	4.8802	65.33	6.2400	6.1652	17.1164	18.7426
PCEG	0.5735	85.0	**1.9526**	3.2559	3.4008	**8.3730**	10.3148	**93.0**	**6.0188**	**5.5957**	23.9041	**6.4153**
PCEG-S	0.4396	85.0	1.9792	3.3881	**3.4199**	10.4026	6.5420	92.33	6.1346	5.8446	**25.8390**	**7.0302**
PCEG-S-DDIM	**0.2654**	81.33	2.0082	3.5066	3.0665	13.3603	**3.4762**	83.0	6.2192	6.1666	20.4415	9.0775



(a) GPMP (b) MPD (c) Diffusion-ES (d) PCEG (e) PCEG-S-DDIM

Fig. 5: Trajectory on unencountered environment of baseline planners. Starting behind the wall, the end effector goes into the lower shelf. Only planners that produced a collision-free path are executed on the real robot. Calibration is used only for visualization when overlaying the executed trajectories on the RGB-D point clouds in (c), (d), and (e).

TABLE IV: Comparison between baselines and our methods in the unseen realistic environment setting over 10 seeds.

Method	Plan$S$[%]↑	Sim$S$[%]↑	Sim$V$↑	Best↓
GPMP	40	0	-	-
MPD	20	0	-	-
Diffusion-ES	40	10	0.0000	15.5475
PCEG	70	60	8.8139	9.5989
PCEG-S-DDIM	100	100	10.0074	10.3066

is improved in both environments. It indicates that PC sampling reduces the likelihood that an individual sample will discover high-quality trajectory modals but enhances global search by restricting the search space for each sample in the informative prior.

E. Computation reduction

Our method can reduce computation by scheduling the number of perturbations:

$$N_p(t) = \max(\text{int}(N_p * (1 - t/T_{\text{diff}})), n), \qquad (19)$$

where $n = 2$ for the point mass, or 5 for 7-DoF manipulator. Early steps emphasize coarse, wide exploration for modal decision; later steps perform narrow, precise refinement to satisfy constraints. This method is referred to as PCEG small (PCEG-S). Furthermore, diffusion planners can be accelerated by DDIM [29]. Focusing on a semantic stage and additional sampling steps, the number of timesteps is non-uniformly reduced to 16. We use full noise to enable PC sampling. With DDIM, $N_p(t)$ adopts $n = 3$ for the point mass, or 6 for the manipulator.

As shown in Table III, PCEG-S lowers computational cost with minimal performance loss. However, PCEG is sensitive to DDIM unlike MPD. Reducing sampling steps limits opportunities to correct deviations from the trained diffusion process caused by GP noise. Despite this sensitivity, PCEG-S-DDIM still attains higher success rate while maintaining comparable path quality and diversity, with runtime on par with MPD-DDIM. In a real-world test on an unseen environment, PCEG-S-DDIM successfully generates feasible trajectories, whereas full MPD struggles to find a solution. Collectively, these results indicate that DDIM does not compromise the generalization capability of our method.

F. Real-world deployment

We conduct real-world experiments with a Franka Research 3 robot to validate the transferability of our model from simulation to real-world settings, as well as its robustness in unseen realistic environments. We design a new scenario where an object must be placed at a target location inside a shelf, with an obstacle blocking the direct route. All diffusion planners are instantiated from a single MPD model pretrained in the Sphere3D environment [11]. We replicate

the real environment in Isaac Gym and compute trajectories in the simulation.

Before execution on the real robot, all planners are tested in Isaac Gym to assess robustness to environment modeling. We report success rate in the planning environment (Plan$S$) and in the simulator (Sim$S$). To quantify diversity, variance is computed over collision-free simulated trajectories (Sim$V$). Best cost is the minimum sum of path smoothness and path length across all generated trajectories. We deploy the trajectory with the best cost in the real world.

As illustrated in Table IV, PCEG methods achieve higher success rate and greater diversity without degrading path quality, indicating superior generalization. Fig. 5 presents qualitative results for each planner. Both GPMP and MPD fail in simulation: GPMP produces dynamically infeasible trajectories, while MPD collides with a novel shelf obstacle. Both Diffusion-ES and PCEG generate valid trajectories, but our method yields qualitatively better paths with reduced upward motion. Even without a manipulator-specific cost, PCEG generates smooth, safe trajectories, demonstrating generalization in the previously unobserved setting.

V. Conclusion

In this paper, we have proposed PCEG, an improved guidance mechanism by integrating local exploration with backward universal guidance and by blending GP noise into the reverse process. This combination increases success rates and trajectory diversity across environments without loss of path quality, and these improvements are generalized to an unseen real-world setup. A promising extension is to combine the proposed method with ensemble-of-costs guidance [26] to search from multiple perspectives, or with movement primitives to enforce smoothness such as B-splines in an advanced MPD [30]. Applying our method as a local planner with onboard sensing is another compelling direction.

References

[1] L. Zhang, S. Wang, K. Cai, Z. Bing, F. Wu, C. Wang, S. Haddadin, and A. Knoll, "APT*: Asymptotically optimal motion planning via adaptively prolated elliptical r-nearest neighbors," *IEEE Robotics and Automation Letters*, vol. 10, no. 10, 2025.

[2] Y. Li, Z. Song, C. Zheng, Z. Bi, K. Chen, M. Y. Wang, and J. Ma, "FRTree planner: Robot navigation in cluttered and unknown environments with tree of free regions," *IEEE Robotics and Automation Letters*, vol. 10, no. 4, 2025.

[3] A. T. Le, K. Pompetzki, J. Carvalho, J. Watson, J. Urain, A. Biess, G. Chalvatzaki, and J. Peters, "Global Tensor Motion Planning," *IEEE Robotics and Automation Letters*, vol. 10, no. 7, 2025.

[4] J. Kuffner and S. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *IEEE International Conference on Robotics and Automation.*, 2000.

[5] M. Mukadam, J. Dong, X. Yan, F. Dellaert, and B. Boots, "Continuous-time gaussian process motion planning via probabilistic inference," *The International Journal of Robotics Research*, vol. 37, no. 11, 2018.

[6] J. Urain, A. T. Le, A. Lambert, G. Chalvatzaki, B. Boots, and J. Peters, "Learning implicit priors for motion optimization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022.

[7] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, "Decision transformer: Reinforcement learning via sequence modeling," *Advances in neural information processing systems*, 2021.

[8] B. Ichter, J. Harrison, and M. Pavone, "Learning sampling distributions for robot motion planning," in *IEEE International Conference on Robotics and Automation*, 2018.

[9] A. Fishman, A. Murali, C. Eppner, B. Peele, B. Boots, and D. Fox, "Motion policy networks," in *conference on Robot Learning*, 2023.

[10] M. Janner, Y. Du, J. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," in *International Conference on Machine Learning*, 2022.

[11] J. Carvalho, A. T. Le, M. Baierl, D. Koert, and J. Peters, "Motion Planning Diffusion: Learning and planning of robot motions with diffusion models," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2023.

[12] P. Dhariwal and A. Nichol, "Diffusion models beat GANs on image synthesis," *Advances in neural information processing systems*, 2021.

[13] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022.

[14] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "STOMP: Stochastic trajectory optimization for motion planning," in *IEEE International Conference on Robotics and Automation*, 2011.

[15] A. Bansal, H.-M. Chu, A. Schwarzschild, S. Sengupta, M. Goldblum, J. Geiping, and T. Goldstein, "Universal guidance for diffusion models," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[16] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the RRT*," in *IEEE International Conference on Robotics and Automation*, 2011.

[17] L. Janson, E. Schmerling, A. Clark, and M. Pavone, "Fast marching tree: A fast marching sampling-based method for optimal motion planning in many dimensions," *The International Journal of Robotics Research*, vol. 34, no. 7, 2015.

[18] J. Xu, K. Song, D. Zhang, H. Dong, Y. Yan, and Q. Meng, "Informed anytime fast marching tree for asymptotically optimal motion planning," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 6, 2020.

[19] S. M. LaValle and J. J. Kuffner Jr, "Randomized kinodynamic planning," *The International Journal of Robotics Research*, vol. 20, no. 5, 2001.

[20] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "CHOMP: Gradient optimization techniques for efficient motion planning," in *IEEE International Conference on Robotics and Automation*, 2009.

[21] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, 2014.

[22] A. Correia and L. A. Alexandre, "A survey of demonstration learning," *Robotics and Autonomous Systems*, vol. 182, 2024.

[23] D. Celestini, D. Gammelli, T. Guffanti, S. D'Amico, E. Capello, and M. Pavone, "Transformer-based model predictive control: Trajectory optimization via sequence modeling," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, 2024.

[24] A. Ajay, Y. Du, A. Gupta, J. Tenenbaum, T. Jaakkola, and P. Agrawal, "Is conditional generative modeling all you need for decision-making?" *arXiv preprint arXiv:2211.15657*, 2022.

[25] B. Yang, H. Su, N. Gkanatsios, T.-W. Ke, A. Jain, J. Schneider, and K. Fragkiadaki, "Diffusion-ES: Gradient-free planning with diffusion for autonomous and instruction-guided driving," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[26] K. Saha, V. Mandadi, J. Reddy, A. Srikanth, A. Agarwal, B. Sen, A. Singh, and M. Krishna, "EDMP: Ensemble-of-costs-guided diffusion for motion planning," in *IEEE International Conference on Robotics and Automation*, 2024.

[27] J. Yu, Y. Wang, C. Zhao, B. Ghanem, and J. Zhang, "FreeDoM: Training-free energy-guided conditional diffusion model," *IEEE/CVF International Conference on Computer Vision*, 2023.

[28] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 2015.

[29] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *International Conference on Learning Representations*, 2021.

[30] J. Carvalho, A. T. Le, P. Kicki, D. Koert, and J. Peters, "Motion planning diffusion: Learning and adapting robot motion planning with diffusion models," *IEEE Transactions on Robotics*, vol. 41, 2025.