# GRACE: Gradient-Free Robot Action Generation via Combined Diffusion-MPPI Posterior Mean Estimation

***Abstract*—Diffusion policies generate multimodal robot action sequences from demonstrations, but steering them toward deployment-time constraints typically relies on differentiable guidance costs. This excludes many practical safety constraints, such as binary collision checks, joint limits, and black-box rollout costs that are nondifferentiable. We propose Gradient-free Robot Action generation via Combined diffusion-MPPI posterior mean Estimation (GRACE), which guides a pretrained diffusion policy with Model Predictive Path Integral (MPPI) control using only forward cost evaluations. Building on the common score-ascent structure of diffusion and MPPI, GRACE constructs a cost-conditioned guidance posterior at each reverse step and estimates its mean with a single MPPI update centered at the diffusion reverse mean. For differentiable costs, GRACE recovers conventional gradient guidance under a first-order, matched-covariance approximation. GRACE attains higher success rates than diffusion-based and sampling-based baselines in simulation. On a real 7-DoF manipulator, GRACE achieves 43/45 successes without obstacle collisions, compared with 17/45 for the unguided prior. Code and experiment videos are available at https://anonymous.4open.science/w/grace-70BB/.**

***Index Terms*—Diffusion policies, MPPI, motion planning, posterior estimation, gradient-free guidance.**

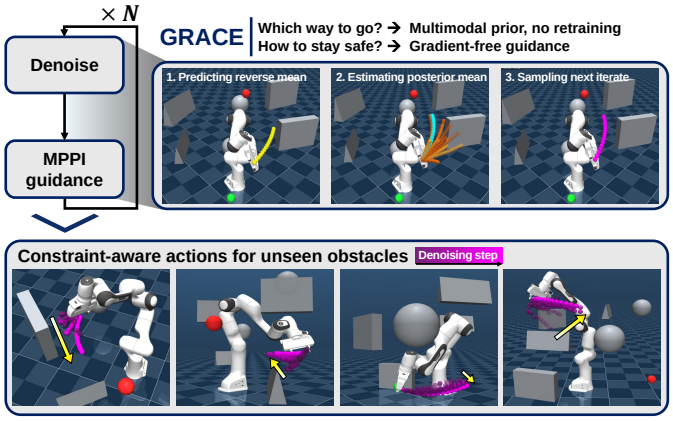



Fig. 1. The multimodal diffusion prior decides which way to go without retraining, while gradient-free guidance decides how to stay safe. At each guided denoising step (top), GRACE predicts the diffusion reverse mean (yellow), estimates the cost-conditioned posterior mean (cyan) from MPPI rollouts (orange), and samples the next iterate (magenta). The bottom panels show different planning scenarios, where trajectories are progressively guided toward constraint satisfaction across denoising steps, visualized from dark to light magenta. Green and red markers denote the start and goal.

## I. INTRODUCTION

ROBOT trajectory generation is a longstanding problem in robotics. An autonomous system must produce trajectories that are dynamically feasible, satisfy task objectives, and remain adaptable to environments that change between deployments. Traditional planners such as RRT [1] explore the state or configuration space to find feasible paths, while trajectory optimizers such as CHOMP [2] refine an initial trajectory by minimizing costs that encode feasibility, smoothness, and task objectives. However, these methods do not explicitly model the multimodal distribution of feasible trajectories that can arise in manipulation and navigation tasks [3].

Generative learning-based methods address this limitation by modeling distributions over trajectories from demonstration data. In particular, diffusion generative models [4] can represent high-dimensional, multimodal distributions through an iterative denoising process and have been applied to robot trajectory generation [5] and visuomotor policy learning [6]. However, sampling from this learned distribution alone does not necessarily ensure that the generated trajectories satisfy additional task objectives or constraints.

Steering the generation process toward such task-specific requirements is the role of guidance. Many deployment-time constraints, such as obstacles that appear only after training, must be imposed at inference time without retraining [7]. Existing gradient-based inference-time methods steer the reverse diffusion process using gradients of task-specific costs evaluated on intermediate denoising samples. MPD [8] uses gradients of differentiable motion-planning costs to bias samples toward low-cost trajectories, while EDMP [9] runs parallel guided denoising processes using an ensemble of collision-cost formulations and guidance schedules. DynaGuide [10] instead derives the guidance gradient from a latent outcome-matching objective evaluated through an external learned dynamics model.

However, these approaches require differentiable guidance objectives and may converge to poor local solutions when the resulting optimization landscape is non-convex [11]. Gradient-free methods remove the differentiability requirement by generating candidate trajectories and scoring them through forward cost evaluations. Diffusion-ES [12] uses an outer evolutionary loop that evaluates clean trajectories with a black-box reward and mutates high-scoring samples through truncated diffusion. JM2D [13] uses Monte Carlo importance sampling to estimate an interaction-weighted joint score over a diffusion plan and a model-based optimization variable. More closely related to our in-loop setting, D-Cubed [14] applies CEM-inspired guidance within reverse diffusion by selecting the lowest-cost latent trajectory for subsequent denoising.

We present Gradient-free Robot Action generation via Combined diffusion-MPPI posterior mean Estimation (GRACE), which formulates each gradient-free reverse-step update as posterior mean estimation over the action trajectory. Motivated by the score-ascent correspondence between diffusion and MPPI [15], [16], GRACE constructs a Boltzmann cost-conditioned posterior around each diffusion reverse mean and estimates its first moment from MPPI-weighted rollouts. Unlike the single-candidate update in D-Cubed, a forward-KL projection of this posterior onto the Gaussian reverse-kernel family yields the posterior first moment as the guided reverse

mean. The resulting reverse kernel is then sampled to obtain the next iterate, as illustrated in Fig. 1. For differentiable costs, the same formulation recovers conventional one-step gradient guidance under a first-order, matched-covariance approximation.

The contributions of this paper are as follows:

- **MPPI-guided posterior-mean diffusion.** GRACE integrates MPPI into reverse diffusion by projecting a Boltzmann cost-conditioned posterior onto the Gaussian reverse-kernel family and estimating its mean from MPPI-weighted rollouts, enabling gradient-free adaptation without retraining.
- **Connection to gradient guidance.** We show that GRACE recovers conventional one-step gradient guidance for differentiable costs under a first-order approximation with the guidance covariance matched to the diffusion reverse covariance.
- **Experimental validation.** GRACE achieves the highest success rates in 2D and 3D simulations, with analyses of trajectory multimodality, guidance sensitivity, and execution horizons, and achieves 43/45 successes without obstacle collisions across three real-robot obstacle conditions.

## II. PRELIMINARIES

### A. Diffusion Models for Trajectory Generation

We adopt the Denoising Diffusion Probabilistic Model (DDPM) formulation [4] for trajectory generation. Let $\mathbf{U}^{(0)} \in \mathbb{R}^{H\times m}$ denote an action sequence of horizon $H$ drawn from a demonstration distribution $p_{\mathrm{data}}$. The forward diffusion process gradually corrupts $\mathbf{U}^{(0)}$ by Gaussian noise over $N$ steps, with $\alpha^{(i)} = 1-\beta^{(i)}$ for a predefined noise schedule $\{\beta^{(i)}\}_{i=1}^{N}$. The $i$-step forward noising distribution can be written in closed form as

$$q(\mathbf{U}^{(i)} \mid \mathbf{U}^{(0)}) = \mathcal{N}\Big(\mathbf{U}^{(i)}; \sqrt{\bar\alpha^{(i)}}\,\mathbf{U}^{(0)},\, (1-\bar\alpha^{(i)})\mathbf{I}\Big), \quad (1)$$

where $\bar\alpha^{(i)} = \prod_{j=1}^{i}\alpha^{(j)}$. Equivalently, it can be reparameterized as

$$\mathbf{U}^{(i)} = \sqrt{\bar\alpha^{(i)}}\,\mathbf{U}^{(0)} + \sqrt{1-\bar\alpha^{(i)}}\,\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I}). \quad (2)$$

Generation reverses this process through a learned Gaussian kernel

$$p_\theta(\mathbf{U}^{(i-1)} \mid \mathbf{U}^{(i)}) = \mathcal{N}\Big(\mathbf{U}^{(i-1)};\, \boldsymbol{\mu}_\theta(\mathbf{U}^{(i)}, i),\, \boldsymbol{\Sigma}^{(i)}\Big), \quad (3)$$

whose mean is parameterized through a learned noise predictor $\boldsymbol{\epsilon}_\theta$ as

$$\boldsymbol{\mu}_\theta(\mathbf{U}^{(i)}, i) = \frac{1}{\sqrt{\alpha^{(i)}}}\left(\mathbf{U}^{(i)} - \frac{\beta^{(i)}}{\sqrt{1-\bar\alpha^{(i)}}}\,\boldsymbol{\epsilon}_\theta(\mathbf{U}^{(i)}, i)\right), \quad (4)$$

trained by minimizing the denoising objective

$$\mathcal{L}_{\mathrm{diff}}(\theta) = \mathbb{E}_{i,\mathbf{U}^{(0)},\boldsymbol{\epsilon}}\Big[\,\big\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{U}^{(i)}, i)\big\|_2^2\,\Big]. \quad (5)$$

### B. Model Predictive Path Integral Control

Consider a discrete-time system $\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{v}_t)$ with state $\mathbf{x}_t \in \mathbb{R}^n$ and applied control $\mathbf{v}_t \in \mathbb{R}^m$. Let $\mathbf{V} = (\mathbf{v}_0, \dots, \mathbf{v}_{H-1})$ denote an applied control sequence. Given $\mathbf{x}_0$, the rollout cost is

$$J(\mathbf{V}) = \sum_{t=0}^{H-1} \ell(\mathbf{x}_t, \mathbf{v}_t) + \phi(\mathbf{x}_H). \quad (6)$$

The information-theoretic formulation [17] yields the optimal control distribution

$$q^\star(\mathbf{V}) = \frac{1}{\eta} e^{-\frac{1}{\lambda}J(\mathbf{V})}\mathcal{N}(\mathbf{V}; \widetilde{\mathbf{U}}, \boldsymbol{\Sigma}), \quad (7)$$

where $\lambda > 0$ is a temperature, $\mathcal{N}(\mathbf{V}; \widetilde{\mathbf{U}}, \boldsymbol{\Sigma})$ is a Gaussian base measure centered at $\widetilde{\mathbf{U}}$, and $\eta$ is the normalizer. The common uncontrolled choice is $\widetilde{\mathbf{U}} = 0$. MPPI approximates $q^\star$ within the Gaussian family $\mathcal{N}(\mathbf{V}; \mathbf{U}, \boldsymbol{\Sigma})$, optimizing only the mean $\mathbf{U}$ while holding the covariance $\boldsymbol{\Sigma}$ fixed, by minimizing the forward-KL divergence. For fixed $\boldsymbol{\Sigma}$, this gives the moment-matching solution $\mathbf{U}^\star = \mathbb{E}_{q^\star}[\mathbf{V}]$.

Substituting (7) into the moment-matching solution yields a normalized weighted expectation under the Gaussian base measure. We then rewrite this expectation under the proposal $\mathcal{N}(\mathbf{V}; \mathbf{U}, \boldsymbol{\Sigma})$ using importance sampling. Since the base and proposal distributions share the covariance $\boldsymbol{\Sigma}$, their density ratio can be absorbed into the cost, giving

$$\mathbf{U}^\star = \frac{\mathbb{E}_{\mathcal{N}(\mathbf{U},\boldsymbol{\Sigma})}\Big[\mathbf{V}\,\mathrm{e}^{-\frac{1}{\lambda}\widetilde{J}(\mathbf{V})}\Big]}{\mathbb{E}_{\mathcal{N}(\mathbf{U},\boldsymbol{\Sigma})}\Big[\mathrm{e}^{-\frac{1}{\lambda}\widetilde{J}(\mathbf{V})}\Big]}, \quad (8)$$

where the augmented cost is

$$\widetilde{J}(\mathbf{V}) = J(\mathbf{V}) + \lambda\sum_{t=0}^{H-1}(\mathbf{u}_t - \widetilde{\mathbf{u}}_t)^\top\boldsymbol{\Sigma}^{-1}\mathbf{v}_t. \quad (9)$$

Using samples ${}^k\mathbf{V} = \mathbf{U} + {}^k\delta\mathbf{U}$ with ${}^k\delta\mathbf{U} \sim \mathcal{N}(0, \boldsymbol{\Sigma})$, the Monte Carlo estimator of (8) is

$$\mathbf{U}^\star = \mathbf{U} + \sum_{k=1}^{K} {}^kw \cdot {}^k\delta\mathbf{U}, \quad {}^kw = \frac{\mathrm{e}^{-\frac{1}{\lambda}\widetilde{J}({}^k\mathbf{V})}}{\sum_{l=1}^{K}\mathrm{e}^{-\frac{1}{\lambda}\widetilde{J}({}^l\mathbf{V})}}. \quad (10)$$

## III. PROPOSED METHOD

In this section, we develop GRACE from the shared score-ascent structure of diffusion and MPPI, which motivates a cost-conditioned posterior whose mean can be estimated through gradient-free MPPI rollouts. The overall inference pipeline is illustrated in Fig. 2.

### A. Diffusion and MPPI as Score Ascent

Differentiating the log of (1) yields

$$\nabla_{\mathbf{U}^{(i)}}\log q(\mathbf{U}^{(i)} \mid \mathbf{U}^{(0)}) = \frac{-\boldsymbol{\epsilon}}{\sqrt{1-\bar\alpha^{(i)}}}, \quad (11)$$

which shows that the noise target $\boldsymbol{\epsilon}$ is proportional to the score of the forward conditional distribution. At the optimum of the denoising objective (5), the noise predictor recovers the

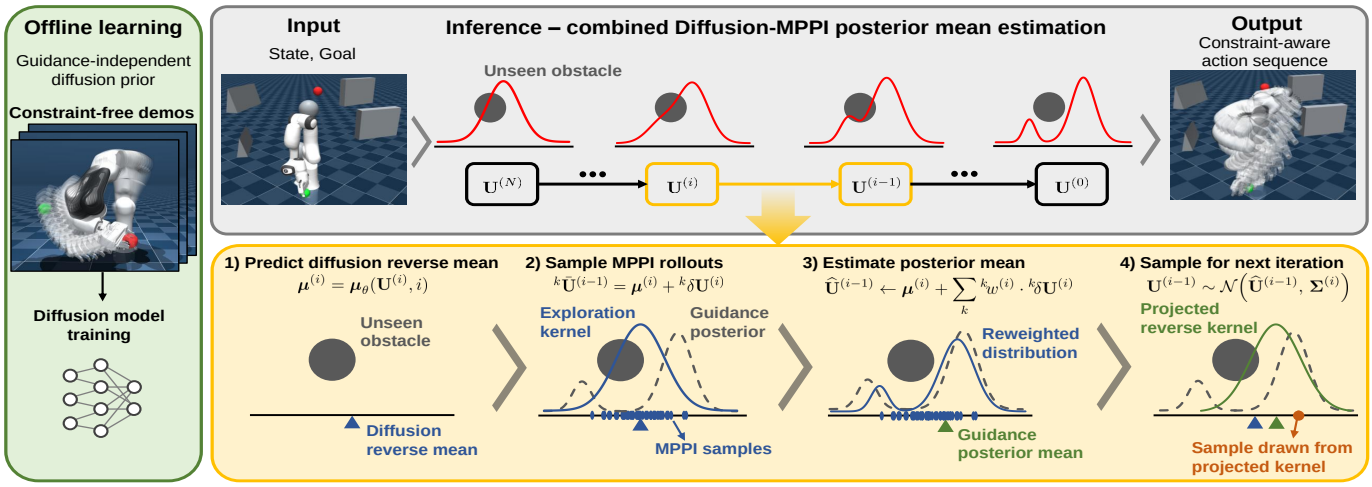



Fig. 2. The GRACE inference pipeline. A diffusion prior learned offline from demonstrations predicts the reverse mean at each denoising step. GRACE samples MPPI rollouts from an exploration kernel centered at this mean and reweights them using deployment-time constraint costs to estimate the guidance posterior mean. The estimated mean defines a projected reverse kernel that retains the diffusion-schedule covariance, from which the next iterate is sampled, enabling constraint-aware action generation without differentiating the cost or the dynamics.

conditional expectation of $\boldsymbol{\epsilon}$ given $\mathbf{U}^{(i)}$. The resulting learned score is

$$\boldsymbol{s}_\theta(\mathbf{U}^{(i)}, i) = -\frac{\boldsymbol{\epsilon}^\star_\theta(\mathbf{U}^{(i)}, i)}{\sqrt{1-\bar{\alpha}^{(i)}}}, \tag{12}$$

which matches $\nabla_{\mathbf{U}^{(i)}} \log q(\mathbf{U}^{(i)})$, where $q(\mathbf{U}^{(i)})$ denotes the forward marginal induced by the data distribution and the forward noising process. Substituting (12) into (4) expresses the reverse mean approximately as a score-ascent step,

$$\boldsymbol{\mu}_\theta(\mathbf{U}^{(i)}, i) \approx \mathbf{U}^{(i)} + \beta^{(i)}\, \boldsymbol{s}_\theta(\mathbf{U}^{(i)}, i). \tag{13}$$

The same view extends to MPPI. Dropping the control prior in (9) reduces the augmented cost to $J$, so the denominator of (8) becomes

$$\Psi(\mathbf{U}) := \mathbb{E}_{\mathbf{V}\sim\mathcal{N}(\mathbf{U},\boldsymbol{\Sigma})}\left[\mathrm{e}^{-\frac{1}{\lambda}J(\mathbf{V})}\right]. \tag{14}$$

Here, $\Psi(\mathbf{U})$ quantifies the expected Boltzmann weight under Gaussian perturbations around $\mathbf{U}$, and its normalized form $\bar{\Psi}(\mathbf{U}) := \Psi(\mathbf{U})/Z$, where $Z := \int \Psi(\mathbf{U})\, d\mathbf{U}$, defines a density over $\mathbf{U}$. Using the Gaussian mean-derivative identity, noting that $e^{-\frac{1}{\lambda}J}$ is independent of $\mathbf{U}$,

$$\nabla_{\mathbf{U}}\Psi(\mathbf{U}) = \boldsymbol{\Sigma}^{-1}\mathbb{E}_{\mathcal{N}(\mathbf{U},\boldsymbol{\Sigma})}\left[(\mathbf{V}-\mathbf{U})\,\mathrm{e}^{-\frac{1}{\lambda}J(\mathbf{V})}\right]. \tag{15}$$

Since $Z$ is constant, $\nabla_{\mathbf{U}} \log \bar{\Psi} = \nabla_{\mathbf{U}}\Psi/\Psi$. Substituting $\mathbf{V} = \mathbf{U} + (\mathbf{V}-\mathbf{U})$ into (8) and using (15) gives

$$\mathbf{U}^\star = \mathbf{U} + \boldsymbol{\Sigma}\, \nabla_{\mathbf{U}} \log \bar{\Psi}(\mathbf{U}). \tag{16}$$

Thus the MPPI update (16) and the approximate diffusion reverse step (13) share the same score-ascent form, each advancing the iterate along a scaled score of an underlying distribution, $q(\mathbf{U}^{(i)})$ for diffusion and $\bar{\Psi}$ for MPPI.

### *B. Cost-Conditioned Guidance Posterior*

The shared score-ascent structure of Section III-A suggests that steering the reverse process toward low-cost trajectories amounts to combining the two ascent directions, which corresponds to targeting the product of the diffusion reverse kernel and a cost-induced likelihood at each step. Bayes' rule provides exactly this product when the deployment-time objective is represented as a conditioning event $\mathcal{C}$ whose likelihood decreases with the rollout cost $J$. The event $\mathcal{C}$ depends only on the action sequence $\mathbf{U}^{(i-1)}$ and is conditionally independent of $\mathbf{U}^{(i)}$ given $\mathbf{U}^{(i-1)}$. Consequently, applying Bayes' rule to the reverse step gives

$$p(\mathbf{U}^{(i-1)} \,|\, \mathbf{U}^{(i)}, \mathcal{C}) \propto p(\mathcal{C} \mid \mathbf{U}^{(i-1)})\, p_\theta(\mathbf{U}^{(i-1)} \mid \mathbf{U}^{(i)}). \tag{17}$$

A Boltzmann likelihood $p(\mathcal{C} \mid \mathbf{U}^{(i-1)}) \propto e^{-\frac{1}{\lambda}J(\mathbf{U}^{(i-1)})}$ is adopted, consistent with the cost-induced target of Section III-A. For compactness, let $\boldsymbol{\mu}^{(i)} := \boldsymbol{\mu}_\theta(\mathbf{U}^{(i)}, i)$ and $p_\theta^{(i)}(\mathbf{U}^{(i-1)}) := p_\theta(\mathbf{U}^{(i-1)} \mid \mathbf{U}^{(i)})$ denote the diffusion reverse mean and the diffusion reverse kernel at step $i$, respectively. Substituting the diffusion reverse kernel (3) into (17) gives the reverse-kernel posterior

$$\pi^{(i)}(\mathbf{U}^{(i-1)}|\mathbf{U}^{(i)}, \mathcal{C}) = \frac{e^{-\frac{1}{\lambda}J(\mathbf{U}^{(i-1)})}\mathcal{N}(\mathbf{U}^{(i-1)}; \boldsymbol{\mu}^{(i)}, \boldsymbol{\Sigma}^{(i)})}{Z^{(i)}}, \tag{18}$$

where $Z^{(i)}$ is the normalizing constant. The reverse-kernel posterior shares the product form of the MPPI optimal distribution (7), with the Gaussian centered at $\boldsymbol{\mu}^{(i)}$, so its mean can be estimated from cost-weighted rollouts sampled from $\mathcal{N}(\boldsymbol{\mu}^{(i)}, \boldsymbol{\Sigma}^{(i)})$. However, as the schedule drives $\boldsymbol{\Sigma}^{(i)} \to \mathbf{0}$ for $i \to 1$, the rollouts concentrate near $\boldsymbol{\mu}^{(i)}$, restricting the range over which cost weighting can shift the posterior mean.

Accordingly, an exploration kernel centered at the diffusion reverse mean and equipped with an independently controlled covariance is introduced:

$$r_g^{(i)}(\mathbf{U}^{(i-1)} \,|\, \mathbf{U}^{(i)}) := \mathcal{N}(\mathbf{U}^{(i-1)}; \boldsymbol{\mu}^{(i)}, \boldsymbol{\Sigma}_g^{(i)}), \tag{19}$$

where $\boldsymbol{\Sigma}_g^{(i)} \succ \mathbf{0}$ is a guidance covariance. When $\boldsymbol{\Sigma}_g^{(i)} - \boldsymbol{\Sigma}^{(i)} \succeq \mathbf{0}$, this exploration kernel is the diffusion reverse kernel convolved with an additional Gaussian exploration kernel,

$$r_g^{(i)}(\mathbf{U}^{(i-1)} | \mathbf{U}^{(i)}) = p_\theta^{(i)}(\mathbf{U}^{(i-1)}) * \mathcal{N}\big(\mathbf{U}^{(i-1)}; \mathbf{0}, \boldsymbol{\Sigma}_g^{(i)} - \boldsymbol{\Sigma}^{(i)}\big), \tag{20}$$

**Algorithm 1** GRACE inference procedure.

**Require:** $\boldsymbol{\epsilon}_\theta$, $\mathbf{x}_0$, $J$, $\{\boldsymbol{\Sigma}^{(i)}, \boldsymbol{\Sigma}_g^{(i)}\}_{i=1}^N$, $K$, $\lambda$, $H_{\text{exec}}$, $N_g$

Sample $\mathbf{U}^{(N)} \sim \mathcal{N}(0, \mathbf{I})$
**for** $i = N, N-1, \ldots, 1$ **do**
  Compute reverse mean $\boldsymbol{\mu}^{(i)} = \boldsymbol{\mu}_\theta(\mathbf{U}^{(i)}, i)$ from (4)
  $\widehat{\mathbf{U}}^{(i-1)} \leftarrow \boldsymbol{\mu}^{(i)}$
  **if** $i \le N_g$ **then**
    **for** $k = 1, \ldots, K$ **do**
      Sample ${}^k\delta\mathbf{U}^{(i)} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_g^{(i)})$
      Form ${}^k\bar{\mathbf{U}}^{(i-1)} = \boldsymbol{\mu}^{(i)} + {}^k\delta\mathbf{U}^{(i)}$
      Roll out and evaluate $J({}^k\bar{\mathbf{U}}^{(i-1)})$
    **end for**
    ${}^k w^{(i)} \leftarrow \dfrac{\mathrm{e}^{-\frac{1}{\lambda}J({}^k\bar{\mathbf{U}}^{(i-1)})}}{\sum_{l=1}^K \mathrm{e}^{-\frac{1}{\lambda}J({}^l\bar{\mathbf{U}}^{(i-1)})}}$
    $\widehat{\mathbf{U}}^{(i-1)} \leftarrow \boldsymbol{\mu}^{(i)} + \sum_{k=1}^K {}^k w^{(i)} \cdot {}^k\delta\mathbf{U}^{(i)}$
  **end if**
  Sample $\mathbf{U}^{(i-1)} \sim \mathcal{N}(\widehat{\mathbf{U}}^{(i-1)}, \boldsymbol{\Sigma}^{(i)})$
**end for**
Execute first $H_{\text{exec}}$ actions of $\mathbf{U}^{(0)}$; replan.

which preserves the diffusion reverse mean while inflating only the spread. Applying Bayes' rule to $r_g^{(i)}(\mathbf{U}^{(i-1)} \,|\, \mathbf{U}^{(i)})$ gives the exploration-kernel posterior, hereafter referred to as the guidance posterior,

$$\rho^{(i)}(\mathbf{U}^{(i-1)}|\mathbf{U}^{(i)}, \mathcal{C}) = \frac{e^{-\frac{1}{\lambda}J(\mathbf{U}^{(i-1)})}\mathcal{N}(\mathbf{U}^{(i-1)}; \boldsymbol{\mu}^{(i)}, \boldsymbol{\Sigma}_g^{(i)})}{Z_g^{(i)}}, \quad (21)$$

where $Z_g^{(i)}$ is the normalizing constant. This is the exact cost-conditioned posterior associated with the exploration kernel $r_g^{(i)}(\mathbf{U}^{(i-1)} \mid \mathbf{U}^{(i)})$. When $\boldsymbol{\Sigma}_g^{(i)} = \boldsymbol{\Sigma}^{(i)}$, the exploration kernel coincides with the diffusion reverse kernel $p_\theta^{(i)}$ and $\rho^{(i)}$ reduces to $\pi^{(i)}$.

### C. KL Projection and MPPI Mean Estimation

Although the exploration-kernel posterior $\rho^{(i)}$ is well defined, the nonlinear and binary terms in $J$ preclude closed-form expressions for its normalizer and moments, and direct sampling from $\rho^{(i)}$ is not readily available. Instead of handling $\rho^{(i)}$ in full, it is projected onto a Gaussian reverse kernel whose mean is free and whose covariance is fixed to $\boldsymbol{\Sigma}^{(i)}$, which reduces the problem to a single moment that admits Monte Carlo estimation. Fixing the covariance to $\boldsymbol{\Sigma}^{(i)}$ ensures that the resulting transition remains in the same schedule-covariance family as the diffusion reverse kernel, while the cost-conditioned information contained in $\rho^{(i)}$ is transferred through the mean.

The corresponding forward-KL projection is

$$\mathbf{U}_g^{\star(i-1)} := \arg\min_{\boldsymbol{\nu}} D_{\mathrm{KL}}\Big(\rho^{(i)} \,\Big\|\, \mathcal{N}\big(\mathbf{U}^{(i-1)}; \boldsymbol{\nu}, \boldsymbol{\Sigma}^{(i)}\big)\Big). \quad (22)$$

Since the covariance of the approximating Gaussian is fixed to $\boldsymbol{\Sigma}^{(i)}$, the forward-KL minimizer is obtained by matching the first moment of $\rho^{(i)}$,

$$\mathbf{U}_g^{\star(i-1)} = \mathbb{E}_{\rho^{(i)}}\left[\mathbf{U}^{(i-1)}\right]. \quad (23)$$

Substituting the exploration-kernel posterior (21) into (23) gives

$$\mathbf{U}_g^{\star(i-1)} = \frac{\mathbb{E}_{\mathbf{U}^{(i-1)} \sim \mathcal{N}\left(\boldsymbol{\mu}^{(i)}, \boldsymbol{\Sigma}_g^{(i)}\right)}\left[\mathbf{U}^{(i-1)} e^{-\frac{1}{\lambda}J(\mathbf{U}^{(i-1)})}\right]}{\mathbb{E}_{\mathbf{U}^{(i-1)} \sim \mathcal{N}\left(\boldsymbol{\mu}^{(i)}, \boldsymbol{\Sigma}_g^{(i)}\right)}\left[e^{-\frac{1}{\lambda}J(\mathbf{U}^{(i-1)})}\right]}. \quad (24)$$

Although this first moment remains unavailable in closed form, both expectations are now taken under the Gaussian $\mathcal{N}(\boldsymbol{\mu}^{(i)}, \boldsymbol{\Sigma}_g^{(i)})$, removing the need to sample from $\rho^{(i)}$ or to evaluate $Z_g^{(i)}$. This ratio has the same form as the MPPI update (8) and is estimated with cost-weighted rollouts. Denoting the resulting Monte Carlo estimate by $\widehat{\mathbf{U}}^{(i-1)}$, consistency of the estimator gives

$$\widehat{\mathbf{U}}^{(i-1)} \longrightarrow \mathbf{U}_g^{\star(i-1)} \qquad \text{as } K \longrightarrow \infty. \quad (25)$$

The estimated mean of the exploration-kernel posterior is then used to define the projected guided reverse kernel

$$q_g^{(i)}\big(\mathbf{U}^{(i-1)} \mid \mathbf{U}^{(i)}, \mathcal{C}\big) := \mathcal{N}\big(\mathbf{U}^{(i-1)}; \widehat{\mathbf{U}}^{(i-1)}, \boldsymbol{\Sigma}^{(i)}\big). \quad (26)$$

In practice, guidance is applied only during the last $N_g$ denoising steps, where the intermediate trajectories are sufficiently denoised for sampled rollouts to provide informative cost evaluations. This also reduces guidance computation by avoiding rollouts at early steps dominated by diffusion noise, following MPD [8]. The overall GRACE reverse process based on this construction is summarized in Algorithm 1.

### D. Gradient Guidance as a Differentiable Special Case

To relate GRACE to gradient-based diffusion guidance, we analyze the first-order behavior of (24) for differentiable costs. Define the deviation from the diffusion reverse mean as $\boldsymbol{\delta} := \mathbf{U}^{(i-1)} - \boldsymbol{\mu}^{(i)}$. Under the exploration kernel, the induced distribution of $\boldsymbol{\delta}$ is $\mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_g^{(i)})$. Let $\nabla J^{(i)} := \nabla_{\mathbf{U}} J(\mathbf{U})|_{\boldsymbol{\mu}^{(i)}}$. The cost is locally approximated as $J(\boldsymbol{\mu}^{(i)} + \boldsymbol{\delta}) \approx J(\boldsymbol{\mu}^{(i)}) + \boldsymbol{\delta}^\top \nabla J^{(i)}$. Substitution into (24) cancels the common factor $e^{-\frac{1}{\lambda}J(\boldsymbol{\mu}^{(i)})}$ and leaves the perturbation exponent

$$-\frac{1}{2}\boldsymbol{\delta}^\top \big(\boldsymbol{\Sigma}_g^{(i)}\big)^{-1}\boldsymbol{\delta} - \frac{1}{\lambda}\boldsymbol{\delta}^\top \nabla J^{(i)}. \quad (27)$$

Completing the square factors the resulting integrand into the Gaussian density $\mathcal{N}(\boldsymbol{\delta}; -\lambda^{-1}\boldsymbol{\Sigma}_g^{(i)}\nabla J^{(i)}, \boldsymbol{\Sigma}_g^{(i)})$ and a constant independent of $\boldsymbol{\delta}$. This constant is common to the numerator and the denominator of (24) and cancels in the ratio, which then reduces to the mean of $\boldsymbol{\delta}$ under the shifted Gaussian.

Since $\mathbf{U}^{(i-1)} = \boldsymbol{\mu}^{(i)} + \boldsymbol{\delta}$, it follows that

$$\mathbf{U}_g^{\star(i-1)} \approx \boldsymbol{\mu}^{(i)} - \frac{1}{\lambda}\boldsymbol{\Sigma}_g^{(i)}\nabla J^{(i)}. \quad (28)$$

Substituting (28) into (26) gives, under the first-order cost approximation and in the infinite-sample limit,

$$\mathbf{U}^{(i-1)} \sim \mathcal{N}\left(\boldsymbol{\mu}^{(i)} - \frac{1}{\lambda}\boldsymbol{\Sigma}_g^{(i)}\nabla J^{(i)}, \boldsymbol{\Sigma}^{(i)}\right). \quad (29)$$

When $\boldsymbol{\Sigma}_g^{(i)} = \boldsymbol{\Sigma}^{(i)}$, (29) reduces to the Taylor-linearized one-step gradient-guided update of [8]. Thus, gradient guidance is recovered as the differentiable, matched-covariance special case of GRACE.

TABLE I
EXPERIMENTAL SETUP FOR THE TWO EVALUATION ENVIRONMENTS.

| | Point-mass | FR3 |
|---|---|---|
| Action | 2D velocity | $\Delta \mathbf{q}$ |
| Prior training data | 10,000 (fixed) | 20,000 (obs.-free) |
| Goal tolerance | 0.05 m | 0.03 rad |
| Step budget | 1,000 | 5,000 |
| Obstacles at inference | fixed + added | 6 random |
| Planning horizon $H$ | 16 | 32 |
| Executed per replan $H_{\text{exec}}$ | 8 | 16 |
| Guidance samples $K$ | 1,000 | 1,000 |
| Guided steps $N_g$ | 10 | 5 |

## IV. EXPERIMENTS

We validate the effectiveness of GRACE on two simulated domains, planar point-mass navigation and 7-DoF Franka Research 3 (FR3) goal-reaching, as well as a real-world experiment with the FR3 manipulator. Complete hyperparameter settings for GRACE and implementation details are available on the anonymous project page at https://anonymous.4open.science/w/grace-70BB/.

### A. Experimental Setup

All experiments were performed on an NVIDIA GeForce RTX 4060 GPU. We use a goal-conditioned Diffusion Policy [6] with either a FiLM-conditioned 1D temporal UNet, denoted **GRACE (UNet)**, or a lightweight FiLM-conditioned 1D residual CNN, denoted **GRACE (CNN)**.

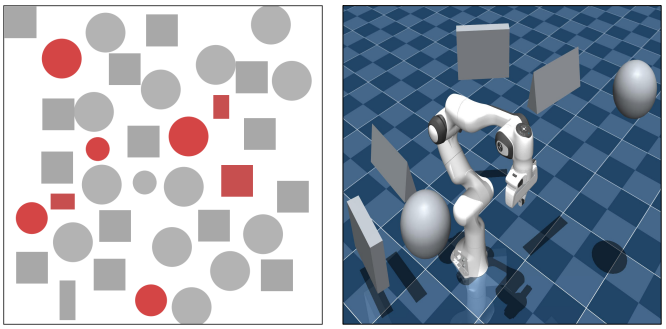
**Environments.** Table I summarizes the setup of the two evaluation environments. The first is the planar point-mass navigation environment of MPD [18], shown in Fig. 3a, where the prior is trained on planner-generated demonstrations collected on a map with fixed obstacles. The start and goal are randomly generated, and additional obstacles absent during training are introduced only at inference time. We conduct three evaluations: navigation under unseen obstacles over 30 random trials, multimodality across 10 start-goal configurations, and sensitivity to $\lambda$, $\Sigma_g$, $K$, and the guidance update. For the multimodality evaluation, we run each method 20 times per configuration and cluster the successful trajectories after uniform arc-length resampling, treating each cluster as one trajectory mode.

The second environment evaluates a 7-DoF manipulator reaching goal joint configurations among 3D obstacles, as shown in Fig. 3b. The prior is trained on obstacle-free feedback-tracking demonstrations, while the obstacles and the start and goal configurations are randomly generated at inference time over 30 trials. For both environments, simulated time does not advance during planning, and a trial succeeds if the system reaches the target within the task-specific tolerance without timeout or safety violations.

**Cost Formulation.** Let $d_o(x_t)$ be the signed distance from state $x_t$ to obstacle $o$, with penetration depth $p_o(x_t) = \max(0, -d_o(x_t))$. We define the obstacle cost over the horizon $H$ as

$$J_{\text{obs}} = \sum_{t=1}^{H} \left( \mathbf{1}_{\text{obs}}(x_t) + \sum_{o \in \mathcal{O}} \left( \frac{p_o(x_t)}{\ell_o} \right)^2 \right), \tag{30}$$

where $\mathbf{1}_{\text{obs}}$ is the exact binary collision indicator used for constraint checking, $\ell_o$ is a per-obstacle normalization length,



(a) (b)

Fig. 3. Simulation environments for the 2D and 3D planning tasks. (a) Planar point-mass navigation, where the red obstacles are absent during training and introduced only at inference time. (b) 7-DoF FR3 manipulator goal-reaching with randomly placed obstacles, all of which are introduced only at inference time.

and the second term is a continuous soft-penetration penalty that provides a gradient signal where the binary term does not. For diffusion-guided methods the goal is supplied by the prior, so the total cost is

$$J = w_{\text{obs}} J_{\text{obs}} + w_{\text{prior}} J_{\text{prior}}, \tag{31}$$

where $J_{\text{prior}}$ penalizes deviation from the diffusion proposal center and is defined as

$$J_{\text{prior}}(\mathbf{V}) = \left\| \mathbf{V} - \boldsymbol{\mu}^{(i)} \right\|_2^2. \tag{32}$$

Since MPPI estimates a cost-weighted posterior mean, constraint costs alone may shift the guided mean toward safe but goal-ineffective trajectories. The prior term regularizes this shift to preserve goal-directed behavior, while the diffusion-free baselines replace $J_{\text{prior}}$ with an explicit goal cost $J_{\text{goal}}$.

In the 2D environment all methods share the same cost in (30) over the exact geometry, so that the comparison isolates the difference between sampling- and gradient-based optimization under identical conditions. In the 3D environment the gradient-guided methods instead operate on a smoothed approximate geometry, whereas the sampling-based methods evaluate the exact geometry through the binary term alone. The 3D setting further adds self-collision, floor contact, and joint-limit constraints, aggregated as $J_{\text{safe}} = w_{\text{obs}} J_{\text{obs}} + w_{\text{self}} J_{\text{self}} + w_{\text{floor}} J_{\text{floor}} + w_{\text{jl}} J_{\text{jl}}$, which replaces $w_{\text{obs}} J_{\text{obs}}$ in (31).

**Baselines.** We compare against three families. The first consists of diffusion-free sampling-based planners, namely **CEM** [19], **MPPI** [17], and **DA-MPPI**, a Diffusion-Annealed MPPI variant inspired by DIAL-MPC [16] that anneals the sampling variance across updates. The second family consists of gradient-based diffusion methods. **Gradient-Guided Diffusion Policy (GG-DP)** applies in-loop refinement during denoising following MPD [8], whereas **Post-Optimization Gradient (PO-Grad)** applies gradient refinement after denoising. The third family consists of sampling-based diffusion methods. **PO-MPPI** applies MPPI refinement after denoising. **Diffusion-ES** [12] applies gradient-free evolutionary search outside the denoising loop. **CEM-DP**, inspired by D-Cubed [14], applies CEM-based guidance within reverse diffusion. All diffusion-based baselines use the same UNet prior as GRACE. All in-loop and post-optimization diffusion guidance

TABLE II
QUANTITATIVE COMPARISON OF PLANNING METHODS IN THE 2D POINT-MASS NAVIGATION TASK.

| Method | Success [%] | Collision Failures | Path Length [m] |
|---|---|---|---|
| CEM | 8/30 (26.7%) | 0/30 | $1.247 \pm 0.143$ |
| MPPI | 8/30 (26.7%) | 0/30 | $1.241 \pm 0.123$ |
| DA-MPPI | 8/30 (26.7%) | 0/30 | $1.233 \pm 0.120$ |
| GG-DP | 18/30 (60.0%) | 12/30 | $2.302 \pm 1.115$ |
| PO-Grad | 18/30 (60.0%) | 12/30 | $2.657 \pm 1.021$ |
| PO-MPPI | 22/30 (73.3%) | 2/30 | $2.546 \pm 1.340$ |
| Diffusion-ES | 7/30 (23.3%) | 23/30 | $2.700 \pm 0.832$ |
| CEM-DP | 15/30 (50.0%) | 15/30 | $1.882 \pm 0.636$ |
| GRACE (CNN) | 25/30 (83.3%) | 2/30 | $3.158 \pm 1.379$ |
| GRACE (UNet) | 25/30 (83.3%) | 1/30 | $2.688 \pm 1.224$ |

TABLE III
SENSITIVITY ANALYSIS OF GRACE IN THE 2D NAVIGATION TASK.

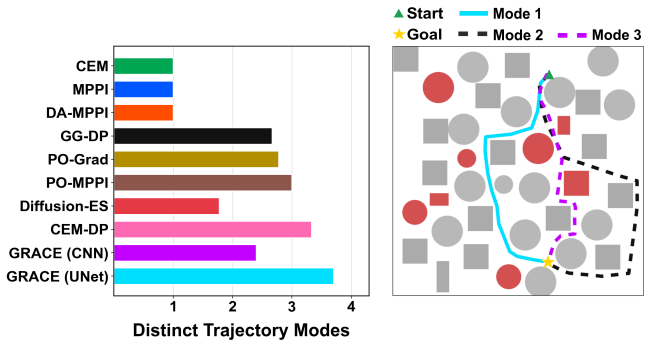
| Update | $\lambda$ | $\Sigma_g$ | $K$ | Success | Collision | Timeout |
|---|---|---|---|---|---|---|
| Weighted | 0.5 | 0.3 | 1,000 | 25/30 | 1 | 4 |
| Weighted | 0.005 | 0.3 | 1,000 | 21/30 | 2 | 7 |
| Weighted | 0.00005 | 0.3 | 1,000 | 22/30 | 7 | 1 |
| Best sample | – | 0.3 | 1,000 | 21/30 | 9 | 0 |
| Weighted | 0.5 | 0.3 | 300 | 22/30 | 2 | 6 |
| Weighted | 0.5 | 0.3 | 50 | 19/30 | 2 | 9 |
| Weighted | 0.5 | 0.1 | 1,000 | 17/30 | 13 | 0 |
| Weighted | 0.5 | 0.7 | 1,000 | 2/30 | 0 | 28 |

methods use the same $N_g$ guidance updates. All diffusion-free sampling-based planners and sampling-based guidance methods use $K = 1{,}000$ samples per update, applying the same control perturbation across the time horizon for each sample, following the single-instance sampling scheme of [20] to improve sample efficiency and rollout smoothness.

### B. Results

**2D navigation performance.** Table II summarizes the planar results. The diffusion-free planners CEM, MPPI, and DA-MPPI all reach $26.7\%$ success with no collisions. Although conservative around obstacles, they lack a learned prior, so goal-cost-driven proposals favor short, near-straight routes over feasible detours and often become trapped in local minima. Their success-only path lengths around $1.24\,\mathrm{m}$ thus reflect easier configurations rather than better planning quality. GG-DP and PO-Grad both reach $60.0\%$ success but incur 12 collision failures, as local gradients can reduce collision cost without directing trajectories toward feasible passages, leading to infeasible refinement in narrow regions.

Among the sampling-based diffusion methods, PO-MPPI reaches $73.3\%$ success with only two collision failures. Compared with PO-Grad, sampling multiple trajectory perturbations rather than following a local gradient enables broader exploration of alternative detours. However, the final diffusion trajectory remains a fixed prior reference throughout post-optimization, penalizing substantial deviations even when needed to avoid unseen obstacles. Diffusion-ES reaches only $23.3\%$ success and incurs 23 collision failures because the outer-loop search relies on complete trajectories generated from the learned prior, limiting adaptation to unseen obstacles. CEM-DP reaches $50.0\%$ success, with all 15 failures due to collision. CEM-DP samples using the diffusion-schedule variance and propagates only the lowest-cost candidate. Decreasing variance restricts exploration, while single-candidate selection discards information from other samples and can steer denoising toward a locally low-cost but collision-prone route. GRACE achieves the highest success rate of $83.3\%$ with both backbones, with two collision failures for CNN and one for UNet. The independently controlled guidance covariance maintains exploration, allowing the cost-weighted update to redirect the trajectory even as the diffusion variance decreases. Moreover, because each guided iterate becomes the next input to the denoiser, the prior can redirect the trajectory toward a different demonstrated detour, rather than deforming a fixed trajectory as in PO-MPPI. GRACE's longer successful paths reflect the detours required to solve harder configurations that other methods fail.




(a) (b)

Fig. 4. Multimodality evaluation in the 2D point-mass navigation task. (a) Average number of trajectory modes produced by each method over the start-goal configurations with at least one successful trajectory, measuring the diversity of generated plans. (b) Trajectory modes produced by GRACE (UNet) for one representative start-goal configuration.

**Trajectory multimodality.** Fig. 4a reports the average number of distinct trajectory modes over configurations with at least one successful trajectory. The diffusion-free planners recover only one mode, consistent with goal-cost-driven proposals favoring direct routes over alternative feasible detours. Among diffusion-based methods, Diffusion-ES shows the lowest diversity with 1.78 modes, which may reflect outer-loop search over complete trajectories rather than guidance during denoising. GRACE (CNN) and GRACE (UNet) instead recover 2.40 and 3.70 modes, respectively. Although each guided reverse transition uses a Gaussian kernel, the projection is conditioned on the current reverse sample, and the next iterate is sampled with the diffusion-schedule covariance. Different reverse trajectories can therefore evolve toward distinct modes rather than collapsing to a single path. Fig. 4b shows the trajectory modes recovered by GRACE (UNet) for one representative start-goal configuration.

**Sensitivity analysis.** Table III summarizes the 2D sensitivity results for temperature $\lambda$, guidance samples $K$, and guidance covariance $\Sigma_g$. Decreasing $\lambda$ concentrates the importance weights, reducing the effective sample size from 619 to 62 and finally to 1, while collision failures increase. The best-sample update incurs 9 collision failures and no timeouts. Unlike best-sample selection, which relies on a single lowest-cost candidate, cost-weighted aggregation incorporates multiple low-cost samples into the guided mean, reducing dependence on any single candidate and yielding fewer collisions. Reducing $K$ lowers success, especially at $K = 50$, where fewer useful rollouts hinder goal progress and increase timeouts. Guidance covariance shows an exploration-convergence trade-off.

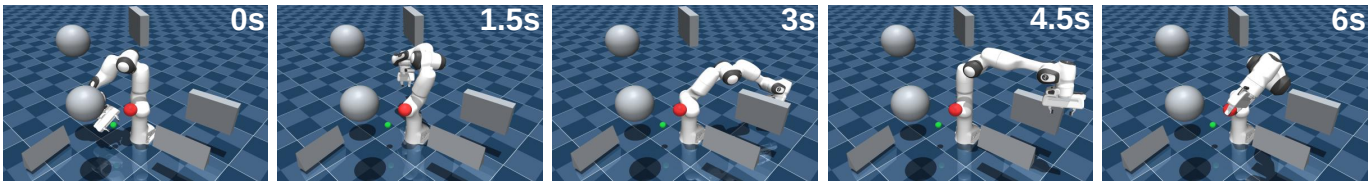



Fig. 5. Representative execution snapshots of GRACE in the 3D FR3 manipulator goal-reaching task. The manipulator reaches the target configuration while satisfying multiple safety constraints.

TABLE IV
QUANTITATIVE COMPARISON OF PLANNING METHODS IN THE 7-DOF FR3 MANIPULATOR GOAL-REACHING TASK.

| Method | Success [%] | Failure Modes | | | | | Computation Time [ms] | Guidance Time [ms] |
|---|---|---|---|---|---|---|---|---|
| | | Obstacle | Self-collision | Joint-limit | Floor | Timeout | | |
| CEM | 24/30 (80.0%) | 1 | 0 | 0 | 0 | 5 | $153.45 \pm 2.55$ | – |
| MPPI | 23/30 (76.7%) | 2 | 0 | 5 | 0 | 0 | $29.88 \pm 0.49$ | – |
| DA-MPPI | 23/30 (76.7%) | 6 | 0 | 0 | 0 | 1 | $154.22 \pm 2.49$ | – |
| GG-DP | 13/30 (43.3%) | 8 | 2 | 3 | 1 | 3 | $1059.03 \pm 7.98$ | $644.57 \pm 4.95$ |
| PO-Grad | 13/30 (43.3%) | 5 | 1 | 6 | 1 | 4 | $956.02 \pm 7.24$ | $546.50 \pm 3.12$ |
| PO-MPPI | 26/30 (86.7%) | 3 | 1 | 0 | 0 | 0 | $629.35 \pm 9.25$ | $181.45 \pm 1.68$ |
| Diffusion-ES | 12/30 (40.0%) | 16 | 0 | 0 | 2 | 0 | $1664.35 \pm 6.72$ | $217.87 \pm 0.70$ |
| CEM-DP | 10/30 (33.3%) | 16 | 1 | 1 | 2 | 0 | $600.97 \pm 16.19$ | $181.55 \pm 1.63$ |
| GRACE (CNN) | 27/30 (90.0%) | 2 | 0 | 0 | 0 | 1 | $258.14 \pm 2.92$ | $181.66 \pm 1.51$ |
| GRACE (UNet) | 28/30 (93.3%) | 1 | 0 | 0 | 0 | 1 | $597.72 \pm 7.87$ | $181.36 \pm 1.18$ |

$\Sigma_g = 0.1$ restricts exploration and causes frequent collisions, whereas $\Sigma_g = 0.7$ avoids collisions but rarely converges due to overly broad perturbations. Intermediate guidance scales better balance safety and convergence.

**3D goal-reaching.** Unlike the 2D task, the prior here is trained on obstacle-free feedback-tracking demonstrations and converges directly to the goal, so performance depends mainly on how the guidance handles the multiple safety constraints. As shown in Table IV, the diffusion-free sampling planners perform well in this setting, with CEM reaching $80.0\%$ success and MPPI and DA-MPPI reaching $76.7\%$. Their failures concentrate on joint limits for MPPI under its fixed exploration variance, obstacle collisions for DA-MPPI as annealing reduces late-stage exploration, and near-goal timeouts for CEM.

GG-DP and PO-Grad both reach $43.3\%$ success with failures across several safety constraints, as combining the safety gradients into a single weighted direction can leave conflicting constraints unresolved, at a substantially higher guidance cost than GRACE. PO-MPPI achieves $86.7\%$ success in this 3D task, suggesting that the limitations of post-update refinement are less pronounced than in 2D navigation, where avoiding unseen obstacles often requires substantial changes to the generated trajectory.

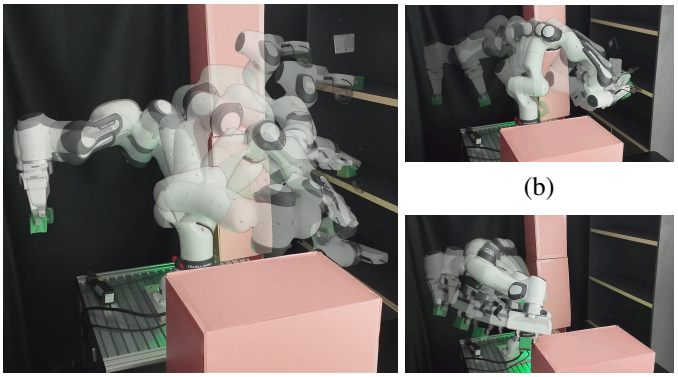
Diffusion-ES and CEM-DP reach only $40.0\%$ and $33.3\%$ success, respectively, each incurring 16 obstacle collisions, consistent with the outer-loop search and single-candidate selection limitations discussed in 2D. GRACE achieves the highest success rates at $90.0\%$ for CNN and $93.3\%$ for UNet, with only two and one obstacle collisions, respectively, indicating that the advantages observed in 2D carry over to multiple simultaneous safety constraints. Its remaining failures arise when nearly all sampled rollouts violate the constraints, leaving no feasible sample to steer the guided mean away from collision, or when no feasible path to the goal is found within the step budget. With the lightweight CNN backbone, the total computation time is further reduced to $258$ ms. PO-MPPI, CEM-DP, and GRACE require nearly identical guidance time of approximately $181$ ms, so these differences arise under comparable sampling costs.

We also evaluated GRACE with different execution horizons $H_{\text{exec}}$. GRACE (UNet) maintained the same $93.3\%$ success rate for $H_{\text{exec}} \in \{8, 16, 32\}$. A larger $H_{\text{exec}}$ reduces replanning frequency but executes a longer trajectory segment before incorporating the updated state. In our static setting, the environment and task constraints remain unchanged during execution, limiting the effect of this reduced adaptivity over the tested range. Fig. 5 shows a representative GRACE (UNet) execution in which the manipulator reaches the target while satisfying the safety constraints in the cluttered environment.

### C. Real-World Experiments

GRACE is validated on a physical 7-DoF FR3 manipulator that places a grasped object on a bookshelf while avoiding obstacles introduced near the shelf at deployment time. The diffusion prior is trained on demonstrations collected in a simulated replica of the workspace without these obstacles, and is deployed without retraining. The action is the joint configuration $\mathbf{q}$ with the gripper command, and the release is produced by the policy rather than triggered externally. On hardware, GRACE uses $H = 32$ and adaptively determines $H_{\text{exec}}$ from the 32-step prediction. The next planning update is triggered before the current chunk is exhausted based on the recent inference time, and the new trajectory replaces the remainder once available. Each trial begins with the object already grasped and targets one of the three shelf levels, with the start configuration and goal placement randomly offset laterally. Three obstacle conditions are tested, with a low obstacle on the left of the shelf, a tall obstacle on the right, and both together. Fig. 6a illustrates this setup. The obstacle, self-collision, and workspace-bound costs are the only terms added at inference time.

Five trials are run per shelf level under each condition, giving 45 trials each for GRACE (UNet) and the unguided diffusion prior (UNet). The unguided prior succeeds in 7, 10, and 0 of the 15 trials under the three conditions, respectively,



(a) (b) (c)

Fig. 6. Real-world experiment. (a) Experimental setup under the third obstacle condition, with both obstacles (pink boxes), goals on all three shelf levels, and laterally randomized start and goal configurations. (b) Successful motion of GRACE and (c) failed motion of the unguided diffusion prior under the same condition, with the bottom shelf as the target.

only where a demonstrated mode happens to remain clear, and fails in every trial once both obstacles are present. GRACE succeeds in 14, 15, and 14 of 15, with no obstacle collision in any trial. The failure under the first condition is a placement inaccuracy, in which the manipulator clears the obstacle but misplaces the object on the shelf, and under the third the MPPI guidance falls into a local minimum between the combined obstacles. Across the three obstacle conditions, GRACE achieves 43/45 successful trials without obstacle collisions, substantially outperforming the unguided prior with 17/45 successes. Figs. 6b and 6c show a trial under the third condition with the bottom shelf as the target, in which GRACE detours around the obstacles and places the object, whereas the unguided prior follows a demonstrated mode blocked by the obstacles and fails.

## V. CONCLUSION

This paper presented GRACE, a gradient-free framework that integrates MPPI-based cost guidance into each diffusion reverse step by shifting the reverse mean while preserving the diffusion-schedule covariance, enabling diffusion policies to incorporate binary, discontinuous, and forward-evaluable deployment-time costs without retraining or backpropagated cost gradients. GRACE achieved the highest success rates in 2D and 3D simulations while producing multiple distinct trajectory modes, and achieved 43/45 successes without obstacle collisions on hardware, compared with 17/45 for the unguided prior. Future work will extend GRACE to broader robotic tasks and develop guidance that steers a single multi-task diffusion prior toward a desired task at inference time without training separate models.

## REFERENCES


[1] S. M. LaValle and J. J. Kuffner, "Rapidly-exploring random trees: Progress and prospects," in *Algorithmic and Computational Robotics: New Directions*, B. R. Donald, K. M. Lynch, and D. Rus, Eds. A K Peters, 2001, pp. 293–308.

[2] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *2009 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2009, pp. 489–494.

[3] T. Osa, "Multimodal trajectory optimization for motion planning," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 983–1001, 2020.

[4] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems (NeurIPS)*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851.

[5] A. Ajay, Y. Du, A. Gupta, J. B. Tenenbaum, T. S. Jaakkola, and P. Agrawal, "Is conditional generative modeling all you need for decision-making?" in *International Conference on Learning Representations (ICLR)*, 2023.

[6] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, vol. 44, no. 10-11, pp. 1684–1704, 2025.

[7] Y. Wang, L. Wang, Y. Du, B. Sundaralingam, X. Yang, Y.-W. Chao, C. Pérez-D'Arpino, D. Fox, and J. Shah, "Inference-time policy steering through human interactions," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 15 626–15 633.

[8] 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, pp. 4881–4901, 2025.

[9] 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 *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 10 351–10 358.

[10] M. Du and S. Song, "Dynaguide: Steering diffusion polices with active dynamic guidance," in *Advances in Neural Information Processing Systems*, vol. 38, 2025, pp. 44 192–44 221.

[11] Y. Luo, C. Sun, J. B. Tenenbaum, and Y. Du, "Potential based diffusion motion planning," *arXiv preprint arXiv:2407.06169*, 2024.

[12] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 15 342–15 353.

[13] W. Jung, U. A. Mishra, N. R. Arachchige, Y. Chen, D. Xu, and S. Kousik, "Joint model-based model-free diffusion for planning with constraints," in *9th Annual Conference on Robot Learning*, 2025.

[14] J. Yamada, S. Zhong, J. Collins, and I. Posner, "D-cubed: Latent diffusion trajectory optimisation for dexterous deformable manipulation," in *Proceedings of The 9th Conference on Robot Learning*, vol. 305. PMLR, 2025, pp. 5039–5055.

[15] Y. Li and M. Chen, "Unifying model predictive path integral control, reinforcement learning, and diffusion models for optimal control and planning," *arXiv preprint arXiv:2502.20476*, 2025.

[16] H. Xue, C. Pan, Z. Yi, G. Qu, and G. Shi, "Full-order sampling-based mpc for torque-level locomotion control via diffusion-style annealing," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 4974–4981.

[17] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Information-theoretic model predictive control: Theory and applications to autonomous driving," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1603–1622, 2018.

[18] 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 *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 1916–1923.

[19] R. Rubinstein, "The cross-entropy method for combinatorial and continuous optimization," *Methodology and computing in applied probability*, vol. 1, no. 2, pp. 127–190, 1999.

[20] D. Kim, E. Im, Y. Kim, M. Lim, and Y. Lee, "Single-instance sampling for computationally efficient and accurate real-time task space mppi control," *IEEE Transactions on Robotics*, vol. 41, pp. 6327–6344, 2025.