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 avoids a deployment-time obstacle that the unguided prior collides with in every trial. Code and experiment videos are available at `https://anonymous.4open.science/w/grace-70BB/`.**

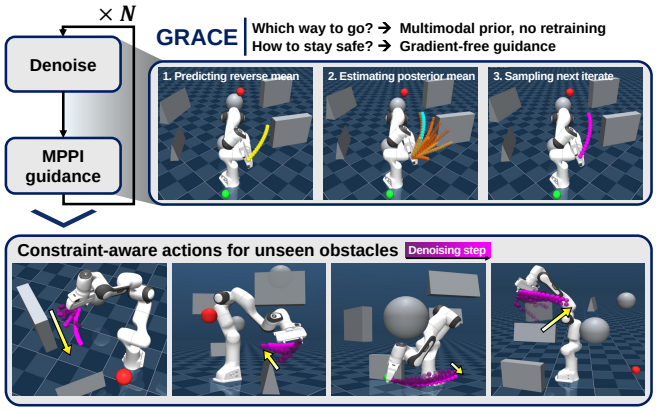


Fig. 1. The multimodal diffusion prior decides which way to go without retraining, while gradient-free guidance decides how to stay safe. At each denoising step (top), GRACE predicts the diffusion reverse mean (yellow), estimates the cost-conditioned posterior mean (cyan) from MPPI rollouts (orange), and samples the next iterate (magenta). Over the denoising process (bottom), the trajectory is refined to avoid unseen obstacles. 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 A* [1] and Rapidly-exploring Random Tree (RRT) [2] explore the state or configuration space to find feasible paths, while trajectory optimizers such as Covariant Hamiltonian Optimization for Motion Planning (CHOMP) [3] and Stochastic Trajectory Optimization for Motion Planning (STOMP) [4] refine an initial trajectory by minimizing costs that encode feasibility, smoothness, and task objectives. These methods have served as the workhorse of deployed robotic systems for decades, but they typically converge to a single mode of the solution space and cannot capture the multiple qualitatively distinct trajectories that are valid in many manipulation and navigation tasks [5].

Generative learning-based methods address this limitation by modeling distributions over trajectories from demonstration data. In particular, diffusion generative models [6] can represent high-dimensional, multimodal distributions through an iterative denoising process and have been applied to robot trajectory generation [7] and visuomotor policy learning [8]. The resulting trajectory prior captures the multimodal structure present in the demonstrations. However, sampling from this learned distribution alone does not necessarily ensure that the generated trajectories satisfy a particular task objective or constraint. Steering the generation process toward such task-specific requirements is the role of guidance.

Guidance can inject the task objective at training time or at inference time. Classifier [9] and classifier-free guidance [10] fold the objective into the prior during training, and therefore require the relevant constraints to be known before the prior is learned. Many deployment-time constraints, such as obstacles that appear only after training, violate this assumption and must instead be imposed at inference time without retraining [11]. Existing inference-time methods steer the reverse diffusion process using gradients of task-specific objectives evaluated on intermediate denoising samples. MPD [12] uses gradients of differentiable motion-planning costs to bias samples toward low-cost trajectories, while EDMP [13] runs parallel guided denoising processes using an ensemble of collision-cost formulations and guidance schedules. DynaGuide [14] 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 [15]. Gradient-free methods remove the differentiability requirement by generating candidate trajectories and scoring them through forward cost evaluations. Diffusion-ES [16] uses an outer evolutionary loop that evaluates clean trajectories with a black-box reward and mutates high-scoring samples through truncated diffusion. JM2D [17] instead jointly de-

noises a plan and a model-based optimization output using a Monte Carlo estimate of their interaction potential.

We present Gradient-free Robot Action generation via Combined diffusion–MPPI posterior mean Estimation (GRACE), a sampling-based guidance framework that integrates MPPI directly into each step of the diffusion denoising process. Motivated by the score-ascent correspondence between diffusion and MPPI [18], [19], GRACE constructs a cost-conditioned guidance posterior around the diffusion reverse mean at each reverse step and estimates its first moment using cost-weighted MPPI rollouts. The estimated posterior mean then parameterizes the mean of a projected reverse kernel, from which the next iterate is sampled, as illustrated in Fig. 1. The resulting method accommodates unseen, potentially nondifferentiable deployment-time constraints without retraining while preserving the stochastic structure of reverse diffusion.

The contributions of this paper are as follows:

- **MPPI-guided reverse diffusion.** GRACE uses the score-ascent correspondence between diffusion and MPPI to incorporate sampling-based guidance into each reverse step, enabling adaptation to unseen deployment-time constraints without retraining.
- **Posterior-mean guidance.** GRACE estimates the mean of a cost-conditioned guidance posterior from cost-weighted rollouts, supporting nondifferentiable costs and recovering one-step gradient guidance under differentiability and matched covariance.
- **Improved planning performance.** GRACE achieves the highest success rates in both simulated tasks, reduces constraint violations relative to gradient-guided diffusion baselines, preserves prior multimodality, and transfers to real hardware.

II. PRELIMINARIES

A. Diffusion Models for Trajectory Generation

We adopt the Denoising Diffusion Probabilistic Model (DDPM) formulation [6] 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_{data}. The forward diffusion process gradually corrupts $\mathbf{U}^{(0)}$ by Gaussian noise over N steps,

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

where $\alpha^{(i)} = 1 - \beta^{(i)}$ for a predefined noise schedule $\{\beta^{(i)}\}_{i=1}^{N}$. The i-step forward noising distribution can be written in closed form as

$$q(\mathbf{U}^{(i)} \mid \mathbf{U}^{(0)}) = \mathcal{N}\Big(\mathbf{U}^{(i)}; \sqrt{\bar\alpha^{(i)}}\,\mathbf{U}^{(0)},\, (1-\bar\alpha^{(i)})\mathbf{I}\Big), \tag{2}$$

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

$$\mathbf{U}^{(i)} = \sqrt{\bar\alpha^{(i)}}\,\mathbf{U}^{(0)} + \sqrt{1-\bar\alpha^{(i)}}\,\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I}). \tag{3}$$

Generation reverses this process through a learned Gaussian kernel

$$p_\theta(\mathbf{U}^{(i-1)} \mid \mathbf{U}^{(i)}) = \mathcal{N}\Big(\mathbf{U}^{(i-1)};\, \boldsymbol{\mu}_\theta(\mathbf{U}^{(i)}, i),\, \boldsymbol{\Sigma}^{(i)}\Big), \tag{4}$$

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

$$\boldsymbol{\mu}_\theta(\mathbf{U}^{(i)}, i) = \frac{1}{\sqrt{\alpha^{(i)}}}\left(\mathbf{U}^{(i)} - \frac{\beta^{(i)}}{\sqrt{1-\bar\alpha^{(i)}}}\,\boldsymbol{\epsilon}_\theta(\mathbf{U}^{(i)}, i)\right), \tag{5}$$

trained by minimizing the denoising objective

$$\mathcal{L}_{\text{diff}}(\theta) = \mathbb{E}_{i, \mathbf{U}^{(0)}, \boldsymbol{\epsilon}}\Big[\, \big\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\mathbf{U}^{(i)}, i)\big\|_2^2 \,\Big]. \tag{6}$$

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 $\mathbf{V} = (\mathbf{v}_0, \ldots, \mathbf{v}_{H-1})$ denote an applied control sequence. Given $\mathbf{x}_0$, the rollout cost is

$$J(\mathbf{V}) = \sum_{t=0}^{H-1} \ell(\mathbf{x}_t, \mathbf{v}_t) + \phi(\mathbf{x}_H). \tag{7}$$

The information-theoretic formulation [20] yields the optimal control distribution

$$q^\star(\mathbf{V}) = \frac{1}{\eta} e^{-\frac{1}{\lambda}J(\mathbf{V})}\mathcal{N}(\mathbf{V}; \widetilde{\mathbf{U}}, \boldsymbol{\Sigma}), \tag{8}$$

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 the normalizer is

$$\eta = \int e^{-\frac{1}{\lambda}J(\mathbf{V})}\mathcal{N}(\mathbf{V}; \widetilde{\mathbf{U}}, \boldsymbol{\Sigma}) d\mathbf{V}. \tag{9}$$

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

$$\mathbf{U}^\star = \arg\min_{\mathbf{U}}\, D_{\text{KL}}(q^\star(\mathbf{V}) \,\|\, \mathcal{N}(\mathbf{V}; \mathbf{U}, \boldsymbol{\Sigma})). \tag{10}$$

For fixed $\boldsymbol{\Sigma}$, this gives the moment-matching solution

$$\mathbf{U}^\star = \mathbb{E}_{q^\star}[\mathbf{V}]. \tag{11}$$

Substituting (8) into (11) and expanding the normalizer using (9) 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

$$\mathbf{U}^\star = \frac{\mathbb{E}_{\mathcal{N}(\mathbf{U}, \boldsymbol{\Sigma})}\Big[\mathbf{V}\, \mathrm{e}^{-\frac{1}{\lambda}\widetilde{J}(\mathbf{V})}\Big]}{\mathbb{E}_{\mathcal{N}(\mathbf{U}, \boldsymbol{\Sigma})}\Big[\mathrm{e}^{-\frac{1}{\lambda}\widetilde{J}(\mathbf{V})}\Big]}, \tag{12}$$

where the augmented cost is

$$\widetilde{J}(\mathbf{V}) = J(\mathbf{V}) + \lambda \sum_{t=0}^{H-1} (\mathbf{u}_t - \widetilde{\mathbf{u}}_t)^\top \boldsymbol{\Sigma}^{-1} \mathbf{v}_t. \tag{13}$$

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 (12) is

$$\mathbf{U}^\star = \mathbf{U} + \sum_{k=1}^{K} {}^k w \cdot {}^k\delta\mathbf{U}, \tag{14}$$

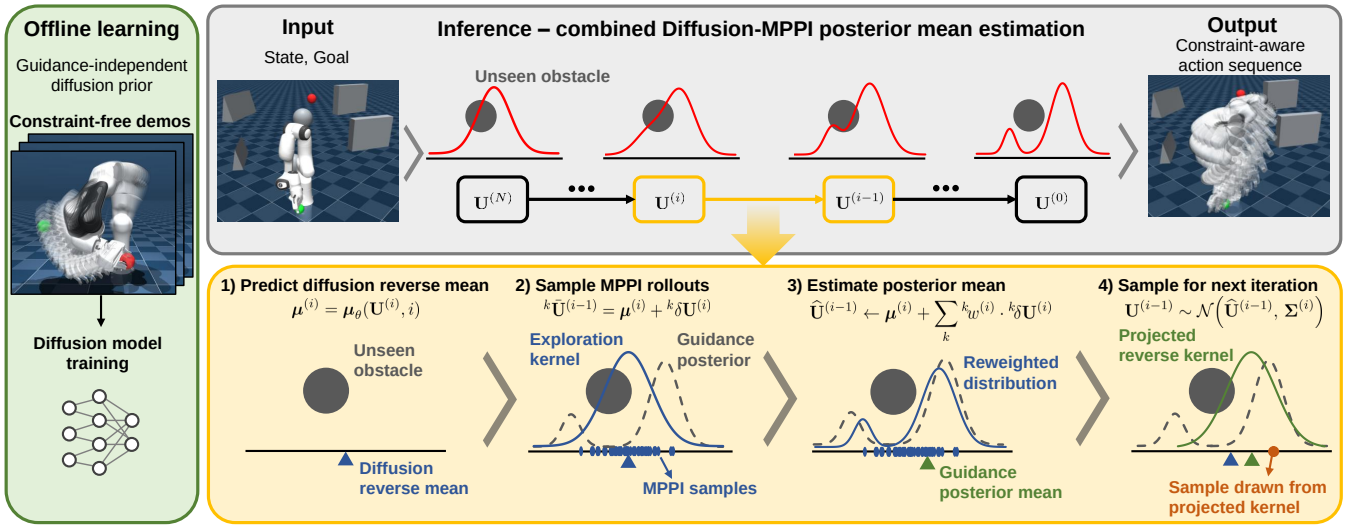


Fig. 2. The GRACE inference pipeline. A diffusion prior learned offline from demonstrations predicts the reverse mean at each denoising step. GRACE samples MPPI rollouts from an exploration kernel centered at this mean and reweights them using deployment-time constraint costs to estimate the guidance 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.

where

$$ {}^{k}w = \frac{\mathrm{e}^{-\frac{1}{\lambda}\widetilde{J}({}^{k}\mathbf{V})}}{\sum_{l=1}^{K}\mathrm{e}^{-\frac{1}{\lambda}\widetilde{J}({}^{l}\mathbf{V})}}. \tag{15} $$

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 (2) yields

$$ \nabla_{\mathbf{U}^{(i)}} \log q(\mathbf{U}^{(i)} \mid \mathbf{U}^{(0)}) = \frac{-\boldsymbol{\epsilon}}{\sqrt{1-\bar{\alpha}^{(i)}}}, \tag{16} $$

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 (6), the noise predictor recovers the conditional expectation of $\boldsymbol{\epsilon}$ given $\mathbf{U}^{(i)}$. The resulting learned score is

$$ \boldsymbol{s}_\theta(\mathbf{U}^{(i)}, i) = -\frac{\boldsymbol{\epsilon}^\star_\theta(\mathbf{U}^{(i)}, i)}{\sqrt{1-\bar{\alpha}^{(i)}}}, \tag{17} $$

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 (17) into (5) expresses the reverse mean approximately as a score-ascent step,

$$ \boldsymbol{\mu}_\theta(\mathbf{U}^{(i)}, i) \approx \mathbf{U}^{(i)} + \beta^{(i)}\, \boldsymbol{s}_\theta(\mathbf{U}^{(i)}, i). \tag{18} $$

The same view extends to MPPI. Dropping the control prior in (13) reduces the augmented cost to J, so the denominator of (12) becomes

$$ \Psi(\mathbf{U}) := \mathbb{E}_{\mathbf{V}\sim\mathcal{N}(\mathbf{U},\boldsymbol{\Sigma})}\left[\mathrm{e}^{-\frac{1}{\lambda}J(\mathbf{V})}\right]. \tag{19} $$

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 $\mathbf{U}$. Substituting $\mathbf{V} = \mathbf{U} + (\mathbf{V} - \mathbf{U})$ into (12) gives

$$ \mathbf{U}^\star = \mathbf{U} + \frac{\mathbb{E}_{\mathcal{N}(\mathbf{U},\boldsymbol{\Sigma})}\left[(\mathbf{V}-\mathbf{U})\,\mathrm{e}^{-\frac{1}{\lambda}J(\mathbf{V})}\right]}{\Psi(\mathbf{U})}. \tag{20} $$

Using the Gaussian identity

$$ \nabla_{\mathbf{U}}\mathcal{N}(\mathbf{V};\mathbf{U},\boldsymbol{\Sigma}) = \boldsymbol{\Sigma}^{-1}(\mathbf{V}-\mathbf{U})\,\mathcal{N}(\mathbf{V};\mathbf{U},\boldsymbol{\Sigma}) \tag{21} $$

and that $e^{-\frac{1}{\lambda}J}$ is independent of $\mathbf{U}$,

$$ \nabla_{\mathbf{U}}\Psi(\mathbf{U}) = \boldsymbol{\Sigma}^{-1}\mathbb{E}_{\mathcal{N}(\mathbf{U},\boldsymbol{\Sigma})}\left[(\mathbf{V}-\mathbf{U})\,\mathrm{e}^{-\frac{1}{\lambda}J(\mathbf{V})}\right]. \tag{22} $$

Since Z is constant, $\nabla_{\mathbf{U}} \log \bar{\Psi} = \nabla_{\mathbf{U}}\Psi/\Psi$. Left-multiplying (22) by $\boldsymbol{\Sigma}$ and dividing by Ψ shows that the second term of (20) equals $\boldsymbol{\Sigma}\nabla_{\mathbf{U}} \log \bar{\Psi}(\mathbf{U})$, so

$$ \mathbf{U}^\star = \mathbf{U} + \boldsymbol{\Sigma}\nabla_{\mathbf{U}} \log \bar{\Psi}(\mathbf{U}). \tag{23} $$

Thus the MPPI update (23) and the approximate diffusion reverse step (18) 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. Applying Bayes' rule to the reverse step gives

$$p(\mathbf{U}^{(i-1)}|\,\mathbf{U}^{(i)},\mathcal{C}) = \frac{p(\mathcal{C}|\mathbf{U}^{(i-1)},\mathbf{U}^{(i)})\,p_\theta(\mathbf{U}^{(i-1)}|\mathbf{U}^{(i)})}{p(\mathcal{C}\,|\,\mathbf{U}^{(i)})}. \quad (24)$$

The event $\mathcal{C}$ depends only on the action sequence $\mathbf{U}^{(i-1)}$ and is conditionally independent of $\mathbf{U}^{(i)}$ given $\mathbf{U}^{(i-1)}$. Furthermore, the denominator is independent of $\mathbf{U}^{(i-1)}$. Consequently,

$$p(\mathbf{U}^{(i-1)}\,|\,\mathbf{U}^{(i)},\mathcal{C}) \propto p(\mathcal{C}\mid \mathbf{U}^{(i-1)})\,p_\theta(\mathbf{U}^{(i-1)}\mid \mathbf{U}^{(i)}). \quad (25)$$

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 (4) into (25) gives the reverse-kernel posterior

$$\pi_i(\mathbf{U}^{(i-1)}|\mathbf{U}^{(i)},\mathcal{C}) = \frac{e^{-\frac{1}{\lambda}J(\mathbf{U}^{(i-1)})}\mathcal{N}(\mathbf{U}^{(i-1)};\boldsymbol{\mu}^{(i)},\boldsymbol{\Sigma}^{(i)})}{Z_i}, \quad (26)$$

where the normalizer is

$$Z_i = \int e^{-\frac{1}{\lambda}J(\mathbf{U}^{(i-1)})}\,\mathcal{N}(\mathbf{U}^{(i-1)};\boldsymbol{\mu}^{(i)},\boldsymbol{\Sigma}^{(i)})\,d\mathbf{U}^{(i-1)}. \quad (27)$$

The reverse-kernel posterior shares the product form of the MPPI optimal distribution (8), 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:

$$r_g^{(i)}(\mathbf{U}^{(i-1)}\,|\,\mathbf{U}^{(i)}) := \mathcal{N}(\mathbf{U}^{(i-1)};\boldsymbol{\mu}^{(i)},\boldsymbol{\Sigma}_g^{(i)}), \quad (28)$$

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,

$$r_g^{(i)}(\mathbf{U}^{(i-1)}|\,\mathbf{U}^{(i)}) = p_\theta^{(i)}(\mathbf{U}^{(i-1)}) * \mathcal{N}\big(\mathbf{U}^{(i-1)};\mathbf{0},\boldsymbol{\Sigma}_g^{(i)} - \boldsymbol{\Sigma}^{(i)}\big), \quad (29)$$

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,

$$\rho_i(\mathbf{U}^{(i-1)}|\mathbf{U}^{(i)},\mathcal{C}) = \frac{e^{-\frac{1}{\lambda}J(\mathbf{U}^{(i-1)})}\mathcal{N}(\mathbf{U}^{(i-1)};\boldsymbol{\mu}^{(i)},\boldsymbol{\Sigma}_g^{(i)})}{Z_g^{(i)}}, \quad (30)$$

where the normalizer is

$$Z_g^{(i)} = \int e^{-\frac{1}{\lambda}J(\mathbf{U}^{(i-1)})}\,\mathcal{N}(\mathbf{U}^{(i-1)};\boldsymbol{\mu}^{(i)},\boldsymbol{\Sigma}_g^{(i)})\,d\mathbf{U}^{(i-1)}. \quad (31)$$

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 ρ_i reduces to π_i.

Algorithm 1 GRACE inference procedure for gradient-free diffusion policy guidance.

Require: denoiser $\boldsymbol{\epsilon}_\theta$; state $\mathbf{x}_0$; cost J; reverse covariances $\{\boldsymbol{\Sigma}^{(i)}\}_{i=1}^N$; guidance covariances $\{\boldsymbol{\Sigma}_g^{(i)}\}_{i=1}^N$; samples K; temperature λ; execution horizon H_{exec}

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 (5)
  for $k = 1, \ldots, K$ **do**
    Sample ${}^k\delta\mathbf{U}^{(i)} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_g^{(i)})$
    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
  ${}^kw^{(i)} \leftarrow \dfrac{\mathrm{e}^{-\frac{1}{\lambda}J({}^k\bar{\mathbf{U}}^{(i-1)})}}{\sum_{l=1}^K \mathrm{e}^{-\frac{1}{\lambda}J({}^l\bar{\mathbf{U}}^{(i-1)})}}$
  $\widehat{\mathbf{U}}^{(i-1)} \leftarrow \boldsymbol{\mu}^{(i)} + \sum_k {}^kw^{(i)} \cdot {}^k\delta\mathbf{U}^{(i)}$
  if $\boldsymbol{\Sigma}^{(i)} \neq \mathbf{0}$ **then**
    Sample $\mathbf{U}^{(i-1)} \sim \mathcal{N}\big(\widehat{\mathbf{U}}^{(i-1)}, \boldsymbol{\Sigma}^{(i)}\big)$
  else
    $\mathbf{U}^{(i-1)} \leftarrow \widehat{\mathbf{U}}^{(i-1)}$
  end if
end for
Execute first H_{exec} actions of $\mathbf{U}^{(0)}$; replan.

C. KL Projection and MPPI Mean Estimation

Although the exploration-kernel posterior ρ_i is well defined, the nonlinear and binary terms in J preclude closed-form expressions for its normalizer and moments, and direct sampling from ρ_i is not readily available. Instead of handling ρ_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 ρ_i is transferred through the mean.

The corresponding forward-KL projection is

$$\mathbf{U}_g^{\star(i-1)} := \arg\min_{\boldsymbol{\nu}} D_{\mathrm{KL}}\Big(\rho_i \,\Big\|\, \mathcal{N}\big(\mathbf{U}^{(i-1)};\boldsymbol{\nu},\boldsymbol{\Sigma}^{(i)}\big)\Big). \quad (32)$$

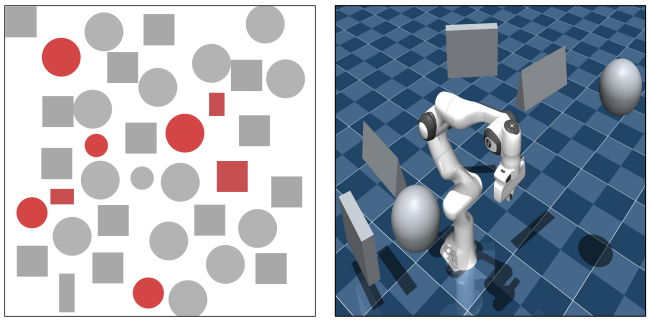
Since the covariance of the approximating Gaussian is fixed to $\boldsymbol{\Sigma}^{(i)}$, the forward-KL minimizer is obtained by matching the first moment of ρ_i,

$$\mathbf{U}_g^{\star(i-1)} = \mathbb{E}_{\rho_i}\left[\mathbf{U}^{(i-1)}\right]. \quad (33)$$

Substituting the exploration-kernel posterior (30) into (33) gives

$$\mathbf{U}_g^{\star(i-1)} = \frac{\mathbb{E}_{\mathbf{U}^{(i-1)}\sim\mathcal{N}\left(\boldsymbol{\mu}^{(i)},\boldsymbol{\Sigma}_g^{(i)}\right)}\left[\mathbf{U}^{(i-1)} e^{-\frac{1}{\lambda}J(\mathbf{U}^{(i-1)})}\right]}{\mathbb{E}_{\mathbf{U}^{(i-1)}\sim\mathcal{N}\left(\boldsymbol{\mu}^{(i)},\boldsymbol{\Sigma}_g^{(i)}\right)}\left[e^{-\frac{1}{\lambda}J(\mathbf{U}^{(i-1)})}\right]}. \quad (34)$$

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 ρ_i or to



(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.

evaluate $Z_g^{(i)}$. This ratio has the same form as the MPPI update (12) and is estimated with cost-weighted rollouts. Denoting the resulting Monte Carlo estimate by $\widehat{\mathbf{U}}^{(i-1)}$, consistency of the estimator gives

$$\widehat{\mathbf{U}}^{(i-1)} \longrightarrow \mathbf{U}_g^{\star(i-1)} \qquad \text{as } K \longrightarrow \infty. \tag{35}$$

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

$$q_g^{(i)}\big(\mathbf{U}^{(i-1)} \mid \mathbf{U}^{(i)}, \mathcal{C}\big) := \mathcal{N}\big(\mathbf{U}^{(i-1)}; \widehat{\mathbf{U}}^{(i-1)}, \boldsymbol{\Sigma}^{(i)}\big). \tag{36}$$

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 (34) 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 (34) cancels the common factor $e^{-\frac{1}{\lambda} J(\boldsymbol{\mu}^{(i)})}$ and leaves the perturbation exponent

$$-\frac{1}{2}\boldsymbol{\delta}^\top \big(\boldsymbol{\Sigma}_g^{(i)}\big)^{-1}\boldsymbol{\delta} - \frac{1}{\lambda}\boldsymbol{\delta}^\top \nabla J_i. \tag{37}$$

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 (34) 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

$$\mathbf{U}_g^{\star(i-1)} \approx \boldsymbol{\mu}^{(i)} - \frac{1}{\lambda}\boldsymbol{\Sigma}_g^{(i)}\nabla J_i. \tag{38}$$

Substituting (38) into (36) gives, under the first-order cost approximation and in the infinite-sample limit,

$$\mathbf{U}^{(i-1)} \sim \mathcal{N}\left(\boldsymbol{\mu}^{(i)} - \frac{1}{\lambda}\boldsymbol{\Sigma}_g^{(i)}\nabla J_i, \boldsymbol{\Sigma}^{(i)}\right). \tag{39}$$

When $\boldsymbol{\Sigma}_g^{(i)} = \boldsymbol{\Sigma}^{(i)}$, (39) reduces to the Taylor-linearized one-step gradient-guided update of [12]. 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)
Planning horizon	16	32
Executed per replan	8	16
Goal tolerance	0.05 m	0.03 rad
Step budget	1,000	5,000
Obstacles at inference	fixed + added	6 random

IV. EXPERIMENTS

In this section, we validate the effectiveness of GRACE on two simulated domains, a planar point-mass navigation task and a goal-reaching task on a 7-DoF Franka Research 3 (FR3) manipulator, as well as on a real-world experiment with the FR3 manipulator. Complete hyperparameter settings for GRACE and all baselines, implementation details, code, and experiment videos 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 [8] 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 [12], 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. In this environment, we conduct two evaluations: feasible navigation under unseen obstacles over 30 random trials, and trajectory multimodality on a fixed episode. For the latter, we cluster 20 successful trajectories per method 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, 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

$$J_{\text{obs}} = \sum_{t=1}^{H}\left(\mathbf{1}_{\text{obs}}(x_t) + \sum_{o\in\mathcal{O}}\left(\frac{p_o(x_t)}{\ell_o}\right)^2\right), \tag{40}$$

where $\mathbf{1}_{\text{obs}}$ is the exact binary collision indicator used for constraint checking, ℓ_o is a per-obstacle normalization length, and the second term is a continuous soft-penetration penalty

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 ± 0.143
MPPI	8/30 (26.7%)	0/30	1.241 ± 0.123
DA-MPPI	8/30 (26.7%)	0/30	1.233 ± 0.120
PO-DP	18/30 (60.0%)	12/30	2.657 ± 1.021
GG-DP	18/30 (60.0%)	12/30	2.302 ± 1.115
Diffusion-ES	7/30 (23.3%)	23/30	2.700 ± 0.832
GRACE (CNN)	20/30 (66.7%)	0/30	3.681 ± 1.659
GRACE (UNet)	25/30 (83.3%)	2/30	3.372 ± 1.538

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

$$J = w_{\text{obs}} J_{\text{obs}} + w_{\text{prior}} J_{\text{prior}}, \tag{41}$$

where J_{prior} penalizes deviation from the prior, and the diffusion-free baselines replace J_{prior} with an explicit goal cost J_{goal}. In the 2D environment all methods share the same cost in (40) 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, which require no gradient, 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 (41).

Baselines. We compare against three families. The first consists of diffusion-free sampling-based planners, namely **MPPI** [20], [21], **CEM** [22], and **DA-MPPI**, a Diffusion-Annealed MPPI variant inspired by DIAL-MPC [19] that anneals the sampling variance across updates. The second family consists of gradient-guided diffusion policies. **Gradient-Guided Diffusion Policy (GG-DP)** applies in-loop refinement during denoising following MPD [12], whereas **Post-Optimized Diffusion Policy (PO-DP)** applies gradient refinement once after denoising. The third family is sampling-based diffusion guidance, represented by **Diffusion-ES** [16], which applies gradient-free evolutionary search outside the denoising loop. All diffusion-based baselines use the same UNet prior as GRACE. Following MPD [12], both GG-DP and GRACE apply guidance only over the final denoising steps, where the reverse mean is informative, rather than at early steps dominated by noise. This reduces guidance computation and is applied identically to both methods for a fair comparison. For all sampling-based planners and the MPPI guidance in GRACE, we apply the same control perturbation across the time horizon for each sample, following the single-instance sampling scheme of [23], which improves sample efficiency and yields smoother rollouts.

B. Results

2D navigation. Table II summarizes the planar results. The diffusion-free planners, CEM, MPPI, and DA-MPPI, all reach 26.7% success with no collisions. They are conservative with respect to obstacles, but lacking any prior they are driven by the goal cost alone, so their uninformed proposal distributions favor short, near-straight routes rather than feasible detours and often become trapped in local minima of the goal cost. Consequently, their success-only path lengths of around 1.24m mainly reflect success on easy configurations rather than better planning quality. PO-DP and GG-DP both reach 60.0% success but incur 12 collision failures. This follows from the local nature of gradient guidance. The collision gradient pushes a colliding segment along the steepest local cost decrease, yet this direction need not lead to a feasible passage. A narrow gap lowers the collision cost locally, so successive gradient steps draw the segment into the gap even when it is too narrow to pass, settling in an infeasible local minimum. Deterministic gradient refinement therefore does not reliably resolve narrow-gap constraint interactions.

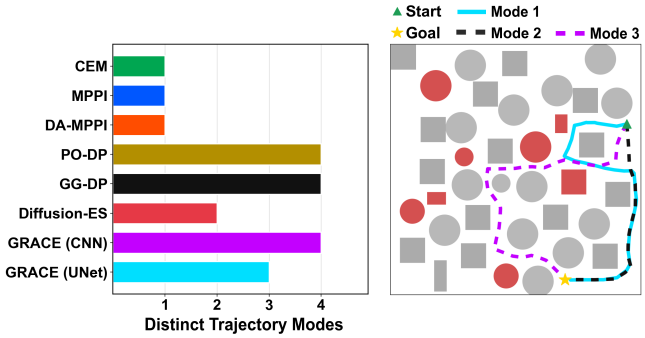


Fig. 4. Multimodality evaluation in the 2D point-mass navigation task. (a) Number of trajectory modes produced by each method, measuring the diversity of generated plans. (b) Representative trajectories of the three modes produced by GRACE (UNet), one per mode.

Diffusion-ES reaches only 23.3% success and incurs the largest number of collision failures at 23, because the cost only scores clean samples in an outer loop while every candidate still comes from the diffusion prior, which was never trained on the unseen obstacles and cannot propose the detours needed to avoid them. GRACE (CNN) reaches 66.7% success with no collisions and GRACE (UNet) 83.3% with only 2 collisions, which occur when every sampled rollout at a reverse step intersects an obstacle and their cost-weighted average therefore remains in collision, a limitation inherent to averaging-based guidance. The GRACE variants have the longest average successful-path lengths because they detour around clutter to solve harder configurations that the other methods fail outright, reflecting broader coverage rather than inefficient planning.

We also assess whether the guidance preserves the multimodality of the prior. Fig. 4a reports the number of distinct trajectory modes produced by each method. The diffusion-free planners show limited diversity and concentrate around a short route, whereas the diffusion-based methods produce multiple detour patterns. For Diffusion-ES, repeated selection and mutation progressively concentrate the population, resulting in fewer recovered modes than the other diffusion-

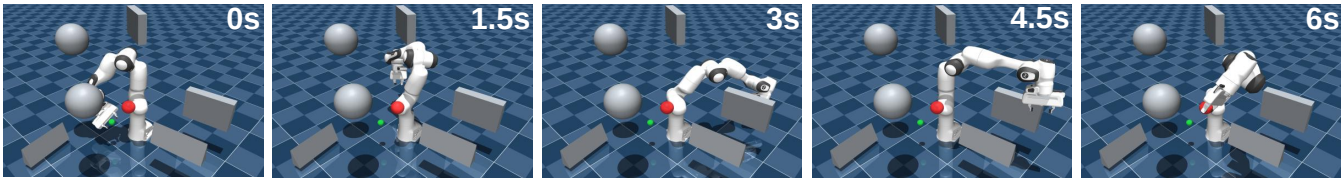


Fig. 5. Representative execution snapshots of GRACE in the 3D FR3 manipulator goal-reaching task. The manipulator reaches the target configuration while satisfying multiple safety constraints.

TABLE III

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 ± 2.55	–
MPPI	23/30 (76.7%)	2	0	5	0	0	29.88 ± 0.49	–
DA-MPPI	23/30 (76.7%)	6	0	0	0	1	154.22 ± 2.49	–
PO-DP	11/30 (36.7%)	6	2	6	2	3	647.30 ± 15.83	220.01 ± 8.76
GG-DP	15/30 (50.0%)	5	2	3	1	4	671.59 ± 9.00	244.72 ± 5.45
Diffusion-ES	12/30 (40.0%)	16	0	0	2	0	1664.35 ± 6.72	217.87 ± 0.70
GRACE (CNN)	27/30 (90.0%)	3	0	0	0	0	179.23 ± 1.91	90.53 ± 0.71
GRACE (UNet)	27/30 (90.0%)	3	0	0	0	0	497.64 ± 7.08	85.88 ± 0.52

based methods. GRACE, in contrast, applies cost-weighted guidance in-loop to shift the reverse mean while retaining the diffusion schedule covariance, allowing it to improve constraint satisfaction without collapsing the prior's multi-modality to a single path. Fig. 4b shows a representative trajectory from each mode recovered by GRACE (UNet).

3D goal-reaching. Unlike the 2D task, where the prior encodes detours that no goal cost recovers, the prior here is trained on obstacle-free feedback-tracking demonstrations and converges directly to the goal, so performance depends on how the guidance handles the multiple safety constraints. As shown in Table III, the diffusion-free sampling planners remain competitive, with CEM reaching 80.0% success and MPPI and DA-MPPI reaching 76.7%, and the failures of each planner concentrate in the mode that matches the sampling update of that planner. MPPI keeps a fixed exploration variance and fails predominantly on joint limits, consistent with the tendency of MPPI to emit aggressive joint-space samples. DA-MPPI anneals this variance, and the failures of DA-MPPI are mostly obstacle collisions, consistent with insufficient late-stage exploration to escape obstacle interactions. CEM refits to elite samples and rarely collides, and the failures of CEM are mostly timeouts from convergence to near-goal plateaus.

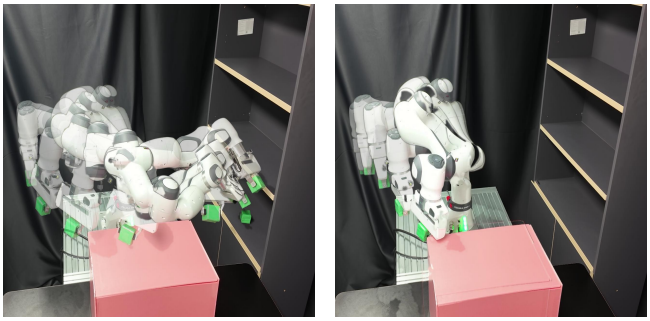
The gradient-guided diffusion policies underperform the diffusion-free sampling planners and GRACE, reaching 36.7% for PO-DP and 50.0% for GG-DP. The gradients of all safety penalties are combined into a single weighted direction, so differences in scale and conflicting directions can cause an update that reduces one penalty to leave others unresolved. Their failures accordingly span all five modes, with self-collision occurring only for these two methods. The higher success of GG-DP over PO-DP suggests that in-loop refinement is more effective than a single post-denoising update, though the guidance of both methods is costly, requiring 220.01 ms and 244.72 ms per step.

Similar to the 2D task, Diffusion-ES struggles to adapt prior trajectories to unseen obstacles, reaching only 40.0% success with by far the most obstacle collisions, and it incurs the largest total computation time at 1664.35 ms from generating and refining multiple complete trajectories. GRACE achieves the best success rate at 90.0% for both backbones, with only three obstacle collisions per backbone, arising from the same averaging limitation as in the 2D task, and no other safety violations. GRACE requires the least guidance computation among all methods, 90.53 ms for GRACE (CNN) and 85.88 ms for GRACE (UNet), as the sampled trajectories are evaluated in parallel through forward rollouts alone. The lower total computation of GRACE (CNN) at 179.23 ms, against 497.64 ms for GRACE (UNet), comes from the faster prediction of the lightweight CNN, which attains the same success rate as the UNet backbone. 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 an obstacle introduced next to the shelf at deployment time. The diffusion prior is trained on demonstrations collected in a simulated replica of the workspace without the obstacle, 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. Each trial begins from a fixed start configuration perturbed by random noise, with the object already grasped, and targets the bottom shelf, where the obstacle most affects the reaching motion. The obstacle, self-collision, and workspace-bound costs are the only terms added at inference time.

Ten trials are run each for GRACE (UNet) and for the unguided diffusion prior (UNet). The unguided prior collides with the obstacle in every trial, following the demonstrated approach that the obstacle now blocks. GRACE succeeds in



(a) (b)

Fig. 6. Real-world experiment. (a) GRACE avoids the obstacle introduced at deployment time (red box) and places the object on the shelf. (b) The unguided diffusion prior collides with the obstacle.

9 of the 10 trials, with no collision in any trial. The single failure is a placement inaccuracy, in which the manipulator clears the obstacle but misplaces the object on the shelf. The guidance thus turns a motion that collides in every trial into one that consistently detours around the obstacle. Fig. 6 visualizes the robot motions of GRACE and the unguided diffusion prior during task execution.

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. This enables diffusion policies to incorporate binary, discontinuous, and forward-evaluable deployment-time costs without retraining or backpropagated cost gradients. Across the 2D and 3D simulation tasks, GRACE achieved the highest success rates while preserving the multimodality of the learned prior, and on the real manipulator it avoided an unseen obstacle in all trials that the unguided prior collided with. These results demonstrate the value of combining learned motion priors with gradient-free online cost evaluation for diffusion-based planning in cluttered, constraint-rich environments. Future work will extend GRACE to a broader range of robotic tasks and investigate a single diffusion prior that captures multiple task behaviors, developing guidance that steers the shared prior toward a desired task at inference time without training separate models.

REFERENCES

[1] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[2] 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.

[3] 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.

[4] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "Stomp: Stochastic trajectory optimization for motion planning," in *2011 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2011, pp. 4569–4574.

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

[6] 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.

[7] 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.

[8] 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.

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

[10] J. Ho and T. Salimans, "Classifier-free diffusion guidance," in *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] G. Williams, A. Aldrich, and E. A. Theodorou, "Model predictive path integral control: From theory to parallel computation," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 344–357, 2017.

[22] 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.

[23] 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.