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

CAPE: Context-Aware Diffusion Policy Via Proximal Mode Expansion for Collision Avoidance

Rui Heng Yang*†, Xuan Zhao*†, Leo Maxime Brunswic†, Montgomery Alban¶
Mateo Clemente†, Tongtong Cao‡, Jun Jin§, Amir Rasouli†

***Abstract*— In robotics, diffusion models can capture multi-modal trajectories from demonstrations, making them a transformative approach in imitation learning. However, achieving optimal performance following this regiment requires a large-scale dataset, which is costly to obtain, especially for challenging tasks, such as collision avoidance. In such tasks, generalization at test time demands coverage of many obstacle types and their spatial configurations, which are impractical to acquire purely via data. Recent works ease this burden with training-free guidance by injecting environmental context at inference, however, it only works when paired with a sufficiently diverse training dataset that yields a conditional trajectory distribution with rich multimodal coverage. To remedy this problem, we propose Context-Aware diffusion policy via Proximal mode Expansion (CAPE), a framework that expands trajectory distribution modes with context-aware prior and guidance at inference via a novel *prior-seeded iterative guided refinement* procedure for motion replanning. The framework generates an initial trajectory plan and executes a short prefix trajectory, and then the remaining trajectory segment is perturbed to an intermediate noise level, forming a context-aware trajectory prior that preserves goal consistency and previously expanded modes. Repeating the process with context-aware guided denoising iteratively expands mode support to allow finding smoother, less collision-prone trajectories. We evaluate CAPE on reaching and pick-and-place tasks in cluttered unseen simulated and real-world settings and show that our proposed approach achieves up to 80% higher success rate and 4× improvement in replanning frequency compared to state-of-the-art, demonstrating better generalization to unseen environments.**

I. Introduction

Diffusion models have achieved remarkable success in generative tasks, such as image synthesis, video generation, and text-to-image translation, owing to their ability to model complex multimodal distributions [1], [2], [3]. Building on this success, recent work has adopted them for robotic control, leveraging their multimodal sampling to model diverse trajectory distributions from demonstrations [4], [5], [6], [7]. Unlike language and vision, where large and standardized datasets enable broad generalization, robotics lacks comparable resources. Demonstrations are typically specific to a platform, task, or environment, making them expensive to collect and even more so to scale. As a result, diffusion models for robot control are typically trained on narrowly distributed datasets, hence struggle to generalize reliably to novel objects, configurations, and real-world scenarios due to insufficient modes for diverse trajectory sampling.

*Equal Contribution. Correspondence to: Rui Heng Yang (rui.heng.yang@huawei.com). †Huawei Technologies Canada. ‡Huawei. §Department of Electrical and Computer Engineering, University of Alberta. ¶Work was done while at Huawei Canada.

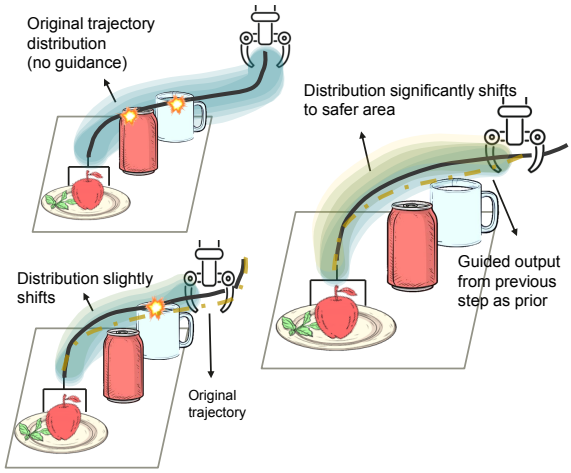

Fig. 1: Overview of the proposed method. Priors derived from previous iterations are incorporated to expand the support of trajectory modes, thereby facilitating the generation of trajectories that are more context-aware.

The aforementioned limitation is particularly significant in challenging scenarios involving collision avoidance, where capturing diverse trajectory modes is essential. A single goal configuration can admit multiple feasible paths depending on obstacle placement and grasp orientation, representing rich trajectory modalities [8]. Capturing the full range of such variations from data alone is impractical: simulation is computationally expensive and prone to sim-to-real gaps, while exhaustive real-world data collection is infeasible. Classical motion planners, such as RRT [9] and reactive methods [10] offer fast, probabilistically complete solutions but rely on fully specified environment models and do not incorporate task-semantic priors from demonstrations, limiting generalization to novel configurations. Diffusion-based planning addresses these limitations by learning rich trajectory distributions from demonstrations, enabling semantically-aware motion generation that can be steered at inference time using available sensor observations.

A common strategy to mitigate limited training data is to apply training-free guidance during inference, steering the diffusion process toward context-aware, task-relevant modes [11], [12]. Guidance leverages the learned trajectory diversity to bias sampling toward modalities underrepresented in training data, such as collision-free trajectories. However, this approach involves a brittle trade-off: weak guidance may be insufficient to prevent unsafe trajectories, while

strong signals risk distorting the learnt distribution, leading to degraded performance and unrealistic trajectories [12], [13].

To address incomplete trajectory modality and the trade-offs inherent in training-free guidance, we propose Context-Aware diffusion policy via Proximal mode Expansion (**CAPE**). CAPE expands mode support iteratively with a context-aware prior and guidance at inference time (Figure 2). After executing a short prefix of the trajectory, CAPE perturbs the remaining segment to an intermediate noise level, constructing a trajectory prior. Using this prior with context-aware guidance, the trajectory modes are expanded to include context-relevant regions. More precisely, the prior preserves the previously expanded mode support and task intent, while the guided refinement process further broadens its distributional support. This procedure yields an increasingly context-aware trajectory distribution, enabling better generalization. For collision avoidance, CAPE yields collision-aware trajectories without the brittle guidance fine-tuning or large-scale data collection. CAPE is also applicable to other contexts, provided that a measurable guidance objective can be defined. Unlike prior-free approaches [11], [14], CAPE couples the prior with training-free guidance for better generalization in unseen environments by producing context-aware and goal-consistent trajectories.

In summary, our contributions are as follows:

- We propose a novel *prior-seeded iterative guided refinement* procedure that constructs a context-aware trajectory prior from the unexecuted plan segments, enabling iterative mode expansion without the need for retraining or large-scale obstacle data collection.
- We conduct theoretical analysis showing that trajectory priors preserve the distribution's anisotropic structure, allowing weaker guidance signals to achieve collision avoidance while avoiding off-distribution collapse.
- We validate the effectiveness of our proposed approach via extensive empirical evaluation on robotic manipulation tasks in simulated and real-world cluttered environments.

II. Related Works

A. Diffusion Models for Robot Control

Diffusion models have gained significant attention in robotics for addressing key challenges in imitation learning, such as multimodal trajectory generation in high-dimensional action spaces [4], [5], [6], [12], [15]. Approaches like Diffusion Behavior Cloning [12], Diffusion Policy [4], and its extension to 3D visual inputs [6] showcase how diffusion can generate full action sequences conditioned on robot observations and environment context. However, limited trajectory diversity in the training data restricts distributional mode support, causing poor generalization in unseen scenarios, particularly cluttered environments with new obstacles.

B. Guidance mechanisms for collision-free diffusion control

Classifier guidance [16] uses pretrained classifiers to steer the denoising process in diffusion-based motion planners for collision avoidance [17], [18], [19]. While effective, this approach requires wide trajectory modality coverage. In fact, APEX [17] collects 500k collision-free trajectories across diverse start-goal configurations and obstacle layouts, ensuring broad workspace coverage for generalization.

Classifier-free guidance (CFG) [20], adopted in [21], [22], interpolates between conditional and unconditional scores. It avoids a separate classifier, but requires evaluating the model twice per diffusion step, increasing latency and computational overhead that may be safety-critical in time-sensitive applications. This method also struggles to handle novel obstacles and novel configurations.

Training-free, loss-based guidance methods [23] apply a cost function directly at inference time, without any additional supervision. MPD [14] leverages this to adapt to novel environments, but its performance is highly sensitive to a single guidance-weight hyperparameter, often requiring environment-specific tuning. RA-DP [11] and Lan-o3dp [24] further omit task constraints from their guidance, resulting in task non-completion when guidance dominates. While such guidance methods help steer the sampling towards underrepresented but safer modes, an improper application degrades performance: overly strong guidance pushes samples off-distribution, leading to poor generalization [12], [13], [8], [25], while weak guidance fails to prevent collisions. Hence, training-free guidance requires sufficient multimodal trajectory distributions to steer sampling toward underrepresented modes while remaining within the distribution [26].

C. Prior-guided Initialization in Diffusion Motion Planning

Prior-guided initialization offers a principled alternative to Gaussian-noise sampling. Denoising Diffusion Bridge Models [27] replace the noise source with structured priors and target distributions, and NaviBridger [28] shows that initializing from an informative action prior improves generalization and downstream performance in visual navigation. In motion planning, READ [29] retrieves context-relevant expert trajectories to define start and goal states. RealDrive [30] interpolates between retrieved demonstrations and current observations to warm-start planning. These retrieval-based designs depend on curated offline data and effective matching, limiting their applicability in novel scenes. A complementary line employs diffusion as a seed trajectory generator. PRESTO [31] and DiffusionSeeder [32] generate trajectory candidates through diffusion and refine them via classical optimization. In contrast to previous motion planning work, CAPE constructs a prior by perturbing its previous output and expands its trajectory mode support with context-aware guidance. CAPE does not require offline retrieval or secondary optimization.

III. Methodology

A. Background

Diffusion-based controllers, such as Diffusion Policy [4], are a probabilistic framework for action generation by iteratively denoising noisy action sequences. We follow the same approach in MPD [14], where the diffusion model generates full trajectories τ instead of action chunks directly,

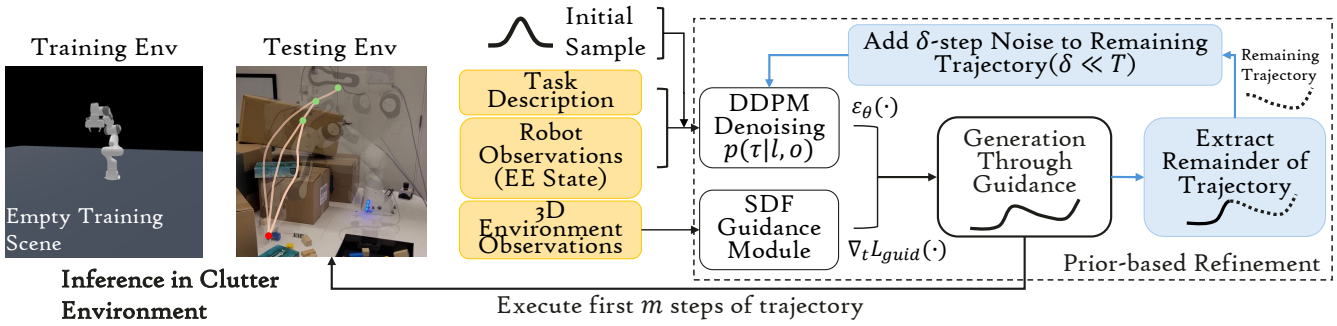

Fig. 2: An overview of the proposed framework: A diffusion model is trained to learn pick-and-place using data from an empty scene. The model uses this skill at inference time in cluttered environments. During inference, the task description and robot observations are sent to the model, and 3D point clouds are used to generate collision-aware guidance signals. **Initial planning**: The noisy trajectory is sampled from Gaussian distribution. **Prior-Seeded Guided Iterative Refinement**: After executing a short prefix trajectory, the remaining trajectory is perturbed with an intermediate noise level δ, forming a prior. The prior preserves task intent and previously expanded mode support, which is further iteratively expanded with collision-aware guidance, until task completion.

capturing both geometric and temporal structures from expert demonstrations. Let $\tau \in \mathbb{R}^{N\times d}$ denote a trajectory consisting of N steps in a d-dimensional action space. We assume to have samples from a conditional distribution of trajectories $p(\tau \mid \mathbf{O})$ to solve a task. The task context $\mathbf{O} = (\ell, o)$ is defined by language instruction or task description ℓ and observations o (e.g. vision, proprioception) of the environment.

We employ a Denoising Diffusion Probabilistic Model (DDPM) [33] to learn the conditional trajectory distribution $p_t(\tau \mid \mathbf{O})$, where the index $t \in \{0, 1, \ldots, T\}$ denotes the noise levels.

The reverse diffusion distribution $q_t(\tau \mid \mathbf{O}; \theta)$ parameterized by θ, constructs a Markov chain starting from $\tau_T \sim \mathcal{N}(0, I)$. At each reverse step, the model denoises by computing

$$\tau_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\tau_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\,\epsilon_\theta(\tau_t, t, \mathbf{O})\right) + \sigma_t z,$$

where $z \sim \mathcal{N}(0, I)$, $\alpha_t = 1 - \beta_t$, $\overline{\alpha}_t = \prod_{k=1}^{t} \alpha_k$, and ϵ_θ denotes the neural network's learned noise predictor. In our work, ϵ_θ is parameterized by a U-Net [34] conditioned on embeddings of $\mathbf{O}$ via cross-attention [35]. The noise model ϵ_θ is trained using a dataset of samples from $p(\tau \mid \mathbf{O})$.

Training-free guidance enables steering the diffusion sampling process during inference without retraining. Specifically, we define a guidance function $\mathcal{L}_{\text{guid}}(\tau, t, \mathbf{O})$ that evaluates a noised trajectory τ_t at noise level t given the task context $\mathbf{O}$. During sampling, this guidance is incorporated via its gradient $\nabla_{\tau_t}\mathcal{L}_{\text{guid}}$, modifying the denoising step as follows:

$$\tau_{t-1} = \frac{1}{\sqrt{\alpha_t}}\left(\tau_t - \frac{1-\alpha_t}{\sqrt{1-\overline{\alpha}_t}}\,\epsilon_\theta(\tau_t, t, \mathbf{O})\right) + \lambda\,\nabla_{\tau_t}\mathcal{L}_{\text{guid}}(\tau_t, t, \mathbf{O}) + \sigma_t z. \quad (1)$$

In our implementation, $\mathcal{L}_{\text{guid}}$ is defined using the signed distance function (SDF) with respect to environment obstacles. This enables us to steer trajectories away from collisions at sampling time, without requiring any additional training [14].

B. Modality Expansion via Prior-Seeded Iterative Guided Refinement Procedure

This section motivates our iterative refinement procedure, which uses priors constructed from previous trajectories to expand the distribution modes for better generalization.

The Guidance-Denoising Tension: A key difficulty for effective training-free guidance is the anisotropy of learned trajectory distributions. Data scarcity concentrates probability mass around limited modes, creating distributions where context-aware trajectories often reside in low-density regions, on the edges of the support. This creates opposing forces: guidance steers sampling toward context-dependent low-density areas while the denoising process pushes noisy samples τ_t toward high-density regions of $p(\tau|\mathbf{O})$. Resolving this conflict requires strong guidance to overcome denoising bias toward familiar trajectories, but excessive guidance from high-magnitude signals or constant injection produces off-distribution samples, leading to poor generalization. Guidance strength scheduling could address this but requires extensive task-specific tuning.

Trajectory-Prior Guided Sampling: Our approach addresses this fundamental tension by employing trajectory priors that preserve the distribution's anisotropic structure while iteratively expanding its mode support in contextually relevant directions, achieving better generalization. Rather than fighting the anisotropy or destroying it with isotropic Gaussian noise, we leverage it by constructing structured priors that guide the expansion process.

We modify the diffusion sampling by first selecting a plausible prior $\widetilde{\tau}_{t=\delta}$ at intermediate noise level $\delta \in [0, T]$, constructed from the unused segment of the previously planned trajectory $\widetilde{\tau}_{t=0}$. Using the remainder trajectory as prior preserves context-aware information relevant to the current scene and task objectives, unlike arbitrary priors that lack this task-specific context. This prior serves as an anchor for proximal modality expansion when combined with context-aware guidance signals.

The key advantage of our prior is preserving the distribu-

tion's anisotropic structure while expanding mode support, concentrating probability density around structured, task-relevant regions.

The conditional likelihood $q_\delta(\widetilde{\tau}_\delta|\mathbf{O},\tau,\lambda)$ of observing the prior given a specific target trajectory τ is high when τ is near the denoised prior. In contrast, the unconditional likelihood $q_\delta(\widetilde{\tau}_\delta|\mathbf{O},\lambda)$ is much smaller since it considers all possible trajectories. This yields a probability ratio greater than 1 in neighborhoods of the denoised prior. This multiplicative scaling preserves anisotropy while concentrating mass around feasible solutions, enabling efficient sampling with weaker guidance signals.

This directly addresses the fundamental tension between guidance and denoising. By initializing from a prior already located within a task-consistent, high-density neighborhood, the denoising process begins close to a feasible solution and requires only local refinement. Guidance therefore needs only to provide small targeted corrections to steer sampling toward underrepresented collision-free modalities, rather than large distributional shifts, which is why weaker guidance suffices and off-distribution collapse is avoided. Additionally, the re-noising step at each iteration introduces controlled stochasticity that may help escape poor local minima from the previous plan, providing complementary robustness across iterations.

Figure 3 illustrates the limitations of applying guidance without a structured prior in motion planning. We examine guidance strength values $\lambda \in \{0.2, 0.5, 1.0\}$, sampling three trajectories for each parameter setting. Weak guidance ($\lambda = 0.2$) fails to sufficiently steer trajectories away from obstacles, leaving samples trapped near collision-prone modes. Medium guidance ($\lambda = 0.5$) creates conflicting gradients around the central obstacle that pull neighboring waypoints in opposite directions, disrupting trajectory coherence while still resulting in collisions. Strong guidance ($\lambda = 1.0$) overwhelms the sampling process, generating highly distorted trajectories with excessive curvature that are kinematically infeasible for robot execution.

C. Algorithms

Unlike previous works, we expand the trajectory distribution modes gradually via *training-free, prior-seeded iterative guided refinement*. We first introduce the notation and the training procedure of our method. We then present the guided denoising for motion planning in Algorithm 1, and describe our two-phase process in Algorithm 2: **(i)** an initial planning pass proposes a trajectory with weak guidance before executing the trajectory prefix of length m; **(ii)** an iterative refinement phase re-noises the remaining segment and applies guided denoising before each subsequent trajectory prefix execution. This process repeats until task completion.

Notation: A trajectory τ is discretized into N end-effector waypoints, each represented by a 9-dimensional pose vector $x = [p, r]$, where $p \in \mathbb{R}^3$ is position and $r \in \mathbb{R}^6$ is the continuous 6D rotation representation for stable and discontinuity-free orientation modeling [36]. We define the task description as $\ell = \{s_s, s_g\}$ for reach, pick and pick-and-place scenarios,

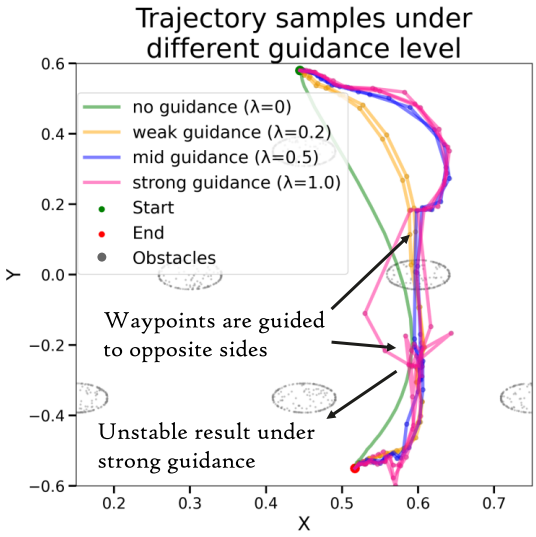

Fig. 3: Trajectory samples under different guidance level in a real planning task without any prior.

where s_s and s_g denote the start and goal states respectively. The observation o comprises robot end-effector states from the previous H time steps, where H is the observation horizon. ℓ and o together form the task context $\mathbf{O}$, guidance strength is $\lambda \geq 0$. The intermediate noise level δ defines the perturbation applied to the unexecuted trajectory suffix forming the prior τ_δ. Guidance is only applied from timestep χ during the guided denoising.

Training: We train on point-to-point motion trajectories of length N collected in obstacle-free scenes, simplifying data collection and reducing cost. This contrasts with previous approaches requiring broad configuration coverage with obstacles [17]. CAPE has no obstacle information during training, so all collision-awareness emerges from inference-time guidance; The framework is dataset-agnostic and also works with obstacle-inclusive datasets. Increasing the multimodality of the initially learned trajectory distribution can yield better performance. Training follows the standard DDPM procedure with the following loss: $\mathcal{L}(\theta) = \|\epsilon - \epsilon_\theta(\tau_t, t)\|_2$

Guided Denoising for Motion Planning: Starting from a noisy trajectory τ_t with noise perturbation level t, we iteratively denoise with the learned model ϵ_θ. For each timestep $t \leq \chi$, we apply the context-aware guidance signal $\lambda \nabla_{\tau_t} \mathcal{L}_{\text{guid}}$ with strength λ. In our collision setting, $\mathcal{L}_{\text{guid}}$ is a collision cost evaluated using the obstacle point cloud $\mathbf{P}_{\text{obs}}$. This guidance gradually expands the relevant distribution modes, steering the samples away from obstacles while preserving task consistency. To ensure task completion, we enforce boundary conditions at every step by clamping the first and last waypoints to the start s_s and goal s_g states. The procedure outputs a collision-aware trajectory τ_0.

Context-Aware Policy via Proximal Mode Expansion: The *initial planning* phase samples a trajectory from a standard Gaussian through guided denoising. The first m steps (prefix trajectory) are executed. The *prior-seeded iterative guided refinement* phase extracts the unexecuted segment, re-noises it to an intermediate noise level $t = \delta$ following the

Algorithm 1: Guided Denoising for Motion Planning

Input : Noisy trajectory τ_t, Noise level t, Guidance start step χ, Task context $\mathbf{O}$, Trained diffusion model ϵ_θ, Obstacle point cloud $\mathbf{P}_{\text{obs}}$, Context-aware guidance function $\mathcal{L}_{\text{guid}}$, Guidance strength λ, Diffusion schedule parameters $(\alpha_t, \bar{\alpha}_t, \sigma_t)$

for $t = t, \ldots, 1$ **do**

  $\mu_t = \frac{1}{\sqrt{\alpha_t}} \left(\tau_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(\tau_t | t, \mathbf{O}) \right)$;

  if $t \leq \chi$ **then**

    `// Apply Cost-Based Guidance`

    $g = -\lambda \nabla_{\tau_{t-1}} \mathcal{L}_{\text{guid}}(\tau_{t-1} = \mu_t, \mathbf{P}_{\text{obs}})$;

    $\tau_{t-1} = \mu_t + g + \sigma_t z$, where $z \sim \mathcal{N}(0, \mathbf{I})$;

  `// Enforce Boundary Constraints`

  $\tau_{t-1}[0] = s_s$, $\tau_{t-1}[H-1] = s_g$;

Output: Denoised trajectory τ_0

Algorithm 2: Collision-Aware Diffusion Policy Via Proximal Mode Expansion

Input : Trained diffusion model ϵ_θ, Task context $\mathbf{O}$, Obstacle point cloud $\mathbf{P}_{\text{obs}}$, Context-aware guidance function $\mathcal{L}_{\text{guid}}$, Guidance strength λ, Perturbation noise level δ, Guidance start step χ, Diffusion schedule parameters $(\alpha_t, \bar{\alpha}_t, \sigma_t)$

Initialize: $\tau_0 \leftarrow$ null, task_done $\leftarrow$ false, $k \leftarrow 0$, first_plan $\leftarrow$ true;

while *not task_done* **do**

  if *first_plan* **then**

    — Initial Planning —;

    $\tau_T[0] = s_s$, $\tau_T[N-1] = s_g$;

    $\tau_T \sim \mathcal{N}(0, \mathbf{I})$;

    $\tau_0^{k=1} \leftarrow$ GuidedDenoising(τ_T, T, χ, $\mathbf{O}$, ϵ_θ, $\mathbf{P}_{\text{obs}}$, $\mathcal{L}_{\text{guid}}$, λ, $(\alpha_t, \bar{\alpha}_t, \sigma_t)$);

    first_plan $\leftarrow$ false;

  else

    — Prior-Seeded Iterative Refinement —;

    Update task context $\mathbf{O}'$ from environment;

    `// Extract remaining trajectory`

    $\widetilde{\tau}_0^k = \text{LinearInterpolate}(s'_s, \tau_0^k[m : N-1], s'_g)$;

    `// Key step: Perturb trajectory to noise level` $t = \delta$

    $\widetilde{\tau}_\delta^k = \sqrt{\bar{\alpha}_t} \widetilde{\tau}_0^k + \sqrt{1-\bar{\alpha}_t} \epsilon$;

    $\tau_0^{k+1} \leftarrow$ GuidedDenoising($\widetilde{\tau}_\delta^k$, δ, t_{start}, $\mathbf{O}'$, ϵ_θ, $\mathbf{P}_{\text{obs}}$, $\mathcal{L}_{\text{guid}}$, λ, $(\alpha_t, \bar{\alpha}_t, \sigma_t)$);

  Execute prefix (first m steps from τ_0^{k+1});

  $k \leftarrow k + 1$;

  if *goal reached or max iterations exceeded* **then**

    task_done $\leftarrow$ true;

forward noising process (line 11 in Algorithm 2), yielding the prior τ_δ. Starting from τ_δ, a new guided denoising pass is executed, augmenting the context-aware modes of the prior distribution. The prior preserves the previously expanded mode support and task consistency, and further iterative expansion is applied on it. At the end of the refinement phase, a new trajectory τ_0 is produced. The controller executes the new trajectory prefix, and phase 2 iterates until task completion.

Collision-Aware Guidance Computation: In our instantiation, the context encodes collision avoidance. Obstacles are represented as a point cloud $\mathbf{P}_{\text{obs}}$ available at inference time only. We approximate the end-effector as a sphere of radius r_{eef}. Given an end-effector position $p \in \mathbb{R}^3$ and a set of obstacles' point clouds, we compute the minimum distance $d(p)$ between the end-effector and the nearest point in $\mathbf{P}_{\text{obs}}$ using the Chamfer distance from PyTorch3D [37]. We define a safety distance of ϵ. $\mathbf{P}_{\text{obs}}$ is only used to generate the training-free guidance during guided denoising. The signed-distance-based guidance cost is then defined as:

$$\mathcal{L}_{\text{guid}}(p) = \begin{cases} -d(p) + (\epsilon + r_{eef}) & \text{if } d(p) \leq \epsilon + r_{eef} \\ 0 & \text{if } d(p) > \epsilon + r_{eef} \end{cases} \quad (2)$$

IV. Experiments

Environment setup. We evaluate CAPE across four progressively challenging collision-avoidance settings to test generalization from the expanded context-aware modes. In pick-and-place tasks, CAPE is responsible for the collision-aware motion planning component, specifically guiding the end-effector trajectory to and from grasp/placement targets while avoiding obstacles. First, a conceptual environment inspired by [38] isolates and visualizes the effects of the structured prior and the iterative guided refinement. Second, realistic simulated tabletop scenes across three difficulty levels – easy, medium, hard (Fig. 4) – evaluate collision avoidance under two observation regimes: full observations (complete obstacle point clouds) and limited observations (wrist-mounted camera). We generate 20 randomized layouts with 5 random initial pose of the robot. To increase the difficulty of the collision-avoidance task, the end-effector height is constrained to remain within 0.3 m above the tabletop so that it needs to move across the obstacles, and the robotic arm must travel a minimum distance of 0.4 m to reach the target object. Finally, we deploy CAPE in real-world cluttered tabletop scenarios, with observations from front-facing and wrist-mounted RGBD cameras. All simulations are run in ManiSkill2 [39]. All experiments are executed on a 7-DoF Franka Panda.

Data generation & training. We collect 1 000 training trajectories using an RRT planner [9] in an obstacle-free simulated environment. We augment the dataset by randomly resampling start points along each trajectory while keeping the target object fixed. Like [14], all trajectories are normalized to a fixed length N using linear interpolation for longer sequences and end-padding for shorter ones. The training set contains no obstacle by design to highlight that CAPE can expand mode support with collision-aware guidance at inference time. CAPE remains compatible with datasets

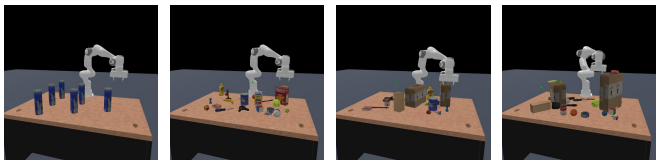

Fig. 4: Simulated environments with increasing level of difficulty used in the experiments. From left to right: 1-conceptual, 2-environment with 25 small obstacles, 3-environment with 15 small and 2 medium size obstacles, and 4- environment with 25 small and 2 large obstacles.

TABLE I: Key hyperparameters used in our experiments.

General Settings		**Model**	
Trajectory Length N	32	Variance Schedule	exponential
State Dimension d	9	Diffusion Steps T	25
Batch Size $\mathcal{B}$	256	Predict Epsilon	True
History Length H	8		
Training		**Inference**	
Learning Rate γ	1e-4	Intermediate Noise Level δ	2
Training Epochs	80	Trajectory Prefix Length m	2
		Guidance Strength λ	0.2
		Guidance Start Step χ	5
Point Cloud Parameters			
Collision Sphere Radius r_{eef}			0.08 m
Safety Margin ϵ			0.06 m

that include obstacles. The same policy is used for both simulation and real-world experiments.

Models. We compare our method against an inference-time guided variant of Diffusion Policy [4] (DP+Guidance), SOTA Motion Planning Diffusion (MPD) [14], which performs one-shot trajectory generation with inference-time guidance, and a variant that adds prior-free refinement to MPD (MPD+Refine) to assess context-aware mode support expansion without the prior. All methods use state-only inputs, the same guidance strength, and the same U-Net backbone [34] as our approach.

Metrics. We report three core metrics: **success rate (SR)**, the fraction of episodes completed without collision; **collision rate (CR)**, the fraction with any contact with obstacles; and **non-completion rate (NCR)**, the fraction that remain collision-free but fail to reach the goal within the horizon. By construction, **SR** + **CR** + **NCR** = 1. We highlight **NCR** to quantify the trade-off between collision avoidance and task completion discussed previously.

A. Experiment in Simulated Environment

DP+Guidance: This method is highly sensitive to the guidance signal, which can dominate the learnt trajectory distribution, suppressing goal-consistent action chunks. This pushes the robot to get trapped in local areas, unable to complete the task as it aim to avoid collisions. Consequently, tasks remain unfinished in most episodes, shown by a NCR $\geq$ 79% and CR $\leq$ 21%.

MPD: On easy scenes with full observation, MPD reaches 96% SR. However, performance degrades all the way to 36% SR with increasing clutter and under partial observability due to missing collision-free mode support. First, sampling trajectories from Gaussian noise provides no collision-awareness, requiring mode expansion from scratch through guidance whose strength is difficult to tune: insufficient guidance results in collisions, while excessive guidance produces unrealistic trajectories. Second, MPD performs one-shot trajectory generation with no further refinement, making it vulnerable to collisions from incomplete observations.

MPD+Refine: MPD+Refine achieves better performance in limited observation scenarios with a 14% SR increase over MPD, as it continuously incorporates up-to-date collision-aware guidance from the environment. However, it suffers from the same limitation as MPD: each refinement starts from Gaussian noise, making it difficult to sample sufficiently diverse trajectories due to limited modal augmentation.

CAPE: Our framework introduces a prior-seeded guided iterative refinement. The trajectory prior preserves the previous context-aware modal augmentations, while guided denoising further expands them with up-to-date guidance. Figure 5 illustrate the importance of using a prior with guided refinement. This yields the best SR, with up to 40% SR gain over MPD and 26% over MPD+Refine. Additionally, the prior provides a significant computational advantage, increasing the refinement frequency by approximately 4×.

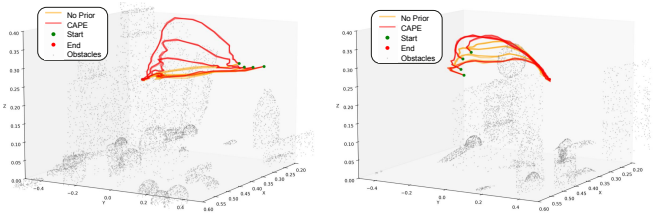

Fig. 5: 3D visualization of trajectory updates during execution in Env4 under full observation. Without a prior, the trajectory is trapped in clutter regions. With a trajectory prior, repeated guided refinement augments context-aware distributional mode support and increases diversity, so the trajectory progressively shifts out of clutter toward the goal.

B. Experiments in Real World

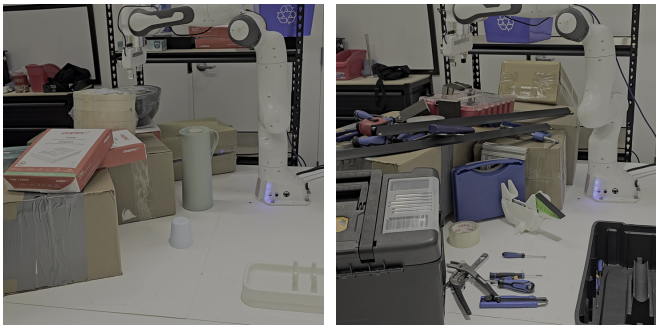

Fig. 6: Real-World Cluttered Environments. Left: Pick-and-Place Cup - The goal is to pick the cup, and place it on the disk rack. Right: Pick Tape - The goal is to pick the tape roll and lift it slightly.

We conducted real-world experiments to benchmark CAPE against MPD and MPD+Refine. We provide quan-

TABLE II: Results of the experiments in the simulated environments. Values are reported as SR(↑)/CR(↓)/NCR(↓). For SR higher value is better and for CR and NCR the lower. Full and Limited refer to the types of observation.

Environments→	Env1: Concept	Env2: Easy	Env3: Medium	Env4: Hard	Env3: Medium	Refine Frequency (Hz)
Policy↓	Full	Full	Full	Full	Limited	
DP+Guidance	0.00/0.00/1.00	0.11/0.00/0.89	0.00/0.12/0.88	0.00/0.19/0.81	0.00/0.21/0.79	N/A
MPD	0.38/0.60/0.02	0.96/0.02/0.02	0.66/0.33/0.01	0.59/0.39/0.02	0.36/0.64/0.00	N/A
MPD+Refine	0.54/0.40/0.06	0.97/0.01/0.02	0.67/0.32/0.01	0.63/0.34/0.03	0.50/0.50/0.00	4.35
CAPE(Ref+Prior)	**0.94**/0.02/0.04	**0.98**/0.02/0.00	**0.82**/0.17/0.01	**0.75**/0.21/0.04	**0.76**/0.24/0.00	**16.67**

TABLE III: Results of the real-world experiments reported as SR(↑)/CR(↓). RF stands for refinement frequency.

Environments→	Pick&Place Cup	Pick Tape	RF (Hz)
MPD	0.80/0.20	0.00/1.00	N/A
MPD+Refine	**1.00/0.00**	0.20/0.80	1.35
CAPE(Ref+Prior)	**1.00/0.00**	**0.80/0.20**	**4.54**

titative results across two environments, running 5 trials for each method in each environment and reporting SR and CR. Table III summarizes our findings. Additionally, we conducted qualitative comparisons between the previous SOTA (MPD) and our method (CAPE), with videos provided in the supplementary material.

MPD: In environments with relatively complete observations and moderate clutter, MPD performs reasonably well, as shown in Environment 1. This aligns with our simulation findings. However, in Environment 2, where obstacles are both more numerous and partially observable, MPD fails and collides with initially unseen obstacles.

MPD+Refine: This augmentation of MPD performs better at handling unseen obstacles. However, since each refinement starts from Gaussian noise, the learned trajectory distribution lacks sufficient collision-aware mode support, resulting in jerky and erratic movements that often lead to collisions. This is reflected in the poor results (SR of 0.2 and CR of 0.8), which aligns with our simulation findings.

CAPE: Our framework achieves the best results. By using a trajectory prior, we can continuously expand the trajectory distribution modes, enabling stronger generalization through more context-aware sampled trajectories. This is evidenced by an 80% improvement over MPD and 60% over MPD+Refine in Pick Tape. However, CAPE has limitations in extreme clutter scenarios where the distributional mode expansion may be insufficient to sample feasible trajectories; the iterative refinement process may fail to adequately expand the trajectory distribution to cover the narrow solution space required for successful navigation.

C. Ablation

Sensitivity to Guidance Strength λ: We compare success rate of MPD, MPD+Refine and CAPE at different guidance strengths. As shown in Figure 7, our method with prior is significantly less sensitive to guidance strength and achieves high success rates at very low guidance levels. The improvement gain from guidance is higher in the more challenging partial observable environment. This is, however, not the case for MPD as the guidance plays a little role to improve the success rate in the absence of the refinement mechanism.

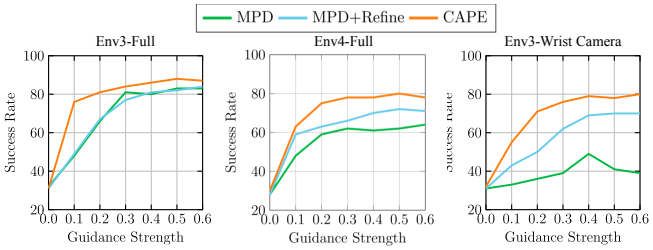

Fig. 7: Impact of guidance strength on SR in the environments under full / wrist camera observation.

Sensitivity to prefix length m and noise level δ: We perform a sweep over the trajectory prefix length m and the intermediate noise level δ. The "Noise" column corresponds to the case where no prior is used (i.e., sampling directly from Gaussian noise). We find that the best performance is achieved with frequent replanning (short prefix) and low noise (small δ), corresponding to rapid denoising from a constantly updating prior. This aligns with intuition: frequent updates allow the prior to incorporate fresh environmental cues, improving responsiveness. In contrast, infrequent replanning increases collision risk due to stale context, while higher noise levels degrade the expanded modes and task information. Detailed results are shown in Table IV. Experiments are conducted in simulation on Environment 3 (medium difficulty) under limited observation conditions.

TABLE IV: Results of the experiments with partial observability. Values are reported as SR(↑)/CR(↓). Noise means no prior.

$m \setminus \delta$	2	3	4	5	6	8	10	Noise
2	**0.76/0.24**	0.72/0.28	0.71/0.29	0.68/0.32	0.61/0.39	0.56/0.44	0.53/0.47	0.53/0.47
3	0.71/0.29	0.70/0.30	0.69/0.31	0.64/0.36	0.66/0.34	0.57/0.43	0.53/0.47	0.50/0.50
4	0.65/0.35	0.69/0.31	0.68/0.32	0.62/0.38	0.58/0.42	0.55/0.45	0.54/0.46	0.51/0.49
5	0.62/0.38	0.61/0.39	0.64/0.36	0.61/0.39	0.60/0.40	0.54/0.46	0.52/0.48	0.52/0.48
8	0.55/0.45	0.56/0.44	0.56/0.44	0.56/0.44	0.52/0.48	0.53/0.47	0.52/0.48	0.49/0.51
10	0.53/0.47	0.55/0.45	0.55/0.45	0.54/0.46	0.53/0.47	0.51/0.49	0.50/0.50	0.50/0.50

Sensitivity to Guidance Start Step χ: We conduct an ablation study to determine the optimal guidance start step, denoted by χ in our denoising algorithm. This parameter governs when context-aware guidance begins during the reverse diffusion process. Starting guidance too late provides insufficient collision avoidance, while applying it too early in the entire denoising can over-correct and generate off-distribution trajectories. With fixed prefix length $m = 2$ and intermediate noise level $\delta = 2$, we find that $\chi = 5$ achieves

optimal performance by balancing collision-awareness with trajectory quality. Table V reports results on Environment 3 (medium difficulty) under limited observation conditions.

TABLE V: Guidance start step χ sweep with guidance strength λ 0.2. Results are reported as SR(↑)/CR(↓).

χ	2	3	4	5	6	7	8	9
SR	0.72/0.28	0.73/0.27	0.73/0.27	**0.76/0.24**	0.75/0.25	0.75/0.25	0.74/0.26	0.74/0.26

V. Conclusion

In this work, we proposed **CAPE**, a novel diffusion-based planning framework that expands trajectory mode support with context-aware prior and guidance at inference via a prior-seeded iterative guided refinement procedure. CAPE addresses a central limitation of diffusion models in robotics: their collapse onto narrow trajectory modes due to limited, task-specific demonstrations. Empirical results across conceptual, simulated, and real-world environments demonstrate that CAPE consistently outperforms state-of-the-art methods, achieving significant improvements in success rate while maintaining trajectory quality in cluttered scenarios.

Despite these advances, our approach has limitations. Our method continuously refines a trajectory prior for context-aware mode expansion; however, if the initial prior is sub-optimal, it may be preferable to reinitialize the planning process entirely. We did not address how to determine prior quality in this work. Additionally, while our guidance approach effectively handles end-effector collision avoidance in the tested scenarios, it does not explicitly ensure full-body collision avoidance. These to be explored in future.

References

[1] D. Epstein, A. Jabri, B. Poole, A. Efros, and A. Holynski, "Diffusion self-guidance for controllable image generation," in *NeurIPS*, 2023.

[2] P. Esser, J. Chiu, P. Atighehchian, J. Granskog, and A. Germanidis, "Structure and content-guided video synthesis with diffusion models," in *ICCV*, 2023.

[3] Y. Li, H. Wang, Q. Jin, J. Hu, P. Chemerys, Y. Fu, Y. Wang, S. Tulyakov, and J. Ren, "Snapfusion: Text-to-image diffusion model on mobile devices within two seconds," in *NeurIPS*, 2023.

[4] 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*, p. 02783649241273668, 2023.

[5] M. Reuss, M. Li, X. Jia, and R. Lioutikov, "Goal conditioned imitation learning using score-based diffusion policies," in *RSS*, 2023.

[6] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations," in *RSS*, 2024.

[7] A. Prasad, K. Lin, J. Wu, L. Zhou, and J. Bohg, "Consistency policy: Accelerated visuomotor policies via consistency distillation," in *RSS*, 2024.

[8] S. Liu, L. Wu, B. Li, H. Tan, H. Chen, Z. Wang, K. Xu, H. Su, and J. Zhu, "RDT-1b: a diffusion foundation model for bimanual manipulation," in *ICLR*, 2025.

[9] S. LAVALLE, "Rapidly-exploring random trees : a new tool for path planning," *Research Report 9811*, 1998.

[10] N. D. Ratliff, J. Issac, D. Kappler, S. Birchfield, and D. Fox, "Riemannian motion policies," 2018. [Online]. Available: https://arxiv.org/abs/1801.02854

[11] X. Ye, R. H. Yang, J. Jin, Y. Li, and A. Rasouli, "Ra-dp: Rapid adaptive diffusion policy for training-free high-frequency robotics replanning," *arXiv preprint arXiv:2503.04051*, 2025.

[12] T. Pearce, T. Rashid, A. Kanervisto, D. Bignell, M. Sun, R. Georgescu, S. V. Macua, S. Z. Tan, I. Momennejad, K. Hofmann, and S. Devlin, "Imitating human behaviour with diffusion models," in *ICLR*, 2023.

[13] Y. Guo, H. Yuan, Y. Yang, M. Chen, and M. Wang, "Gradient guidance for diffusion models: An optimization perspective," in *NeurIPS*, 2024.

[14] 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*, pp. 1–20, 2025.

[15] M. Janner, Y. Du, J. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," in *ICML*, 2022.

[16] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," in *NeurIPS*, 2021.

[17] A. Dastider, H. Fang, and M. Lin, "Apex: Ambidextrous dual-arm robotic manipulation using collision-free generative diffusion models," in *IROS*, 2024.

[18] Y. Zheng, R. Liang, K. ZHENG, J. Zheng, L. Mao, J. Li, W. Gu, R. Ai, S. E. Li, X. Zhan, and J. Liu, "Diffusion-based planning for autonomous driving with flexible guidance," in *ICLR*, 2025.

[19] H. Lin, X. Huang, T. Phan, D. Hayden, H. Zhang, D. Zhao, S. Srinivasa, E. Wolff, and H. Chen, "Causal composition diffusion model for closed-loop traffic generation," in *CVPR*, 2025.

[20] J. Ho and T. Salimans, "Classifier-free diffusion guidance," in *NeurIPSW*, 2021.

[21] Y. Luo, C. Sun, J. B. Tenenbaum, and Y. Du, "Potential based diffusion motion planning," in *ICML*, 2024.

[22] W. Yu, J. Peng, H. Yang, J. Zhang, Y. Duan, J. Ji, and Y. Zhang, "Ldp: A local diffusion planner for efficient robot navigation and collision avoidance," in *IROS*, 2024.

[23] Y. Shen, X. Jiang, Y. Yang, Y. Wang, D. Han, and D. Li, "Understanding and improving training-free loss-based diffusion guidance," in *NeurIPS*, 2024.

[24] Q. Feng, H. Li, Z. Zheng, J. Feng, and A. Knoll, "Language-guided object-centric diffusion policy for collision-aware robotic manipulation," in *ICRA*, 2025.

[25] P. M. Julbe, J. Nubert, H. Hose, S. Trimpe, and K. J. Kuchenbecker, "Diffusion-based approximate mpc: Fast and consistent imitation of multi-modal action distributions," *arXiv preprint arXiv:2504.04603*, 2025.

[26] L. Mao, H. Xu, X. Zhan, W. Zhang, and A. Zhang, "Diffusion-dice: In-sample diffusion guidance for offline reinforcement learning," in *NeurIPS*, 2024.

[27] L. Zhou, A. Lou, S. Khanna, and S. Ermon, in *ICLR*, B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, Eds., 2024.

[28] H. Ren, Y. Zeng, Z. Bi, Z. Wan, J. Huang, and H. Cheng, "Prior does matter: Visual navigation via denoising diffusion bridge models," in *CVPR*, 2025.

[29] T. Oba, M. Walter, and N. Ukita, "Read: Retrieval-enhanced asymmetric diffusion for motion planning," in *CVPR*, 2024.

[30] W. Ding, S. Veer, Y. Chen, Y. Cao, C. Xiao, and M. Pavone, "Realdrive: Retrieval-augmented driving with diffusion models," *arXiv preprint arXiv:2505.24808*, 2025.

[31] M. Seo, Y. Cho, Y. Sung, P. Stone, Y. Zhu, and B. Kim, "Presto: Fast motion planning using diffusion models based on key-configuration environment representation," in *ICRA*, 2025.

[32] H. Huang, B. Sundaralingam, A. Mousavian, A. Murali, K. Goldberg, and D. Fox, "Diffusionseeder: Seeding motion optimization with diffusion for rapid motion planning," in *CoRL*, 2024.

[33] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *NeurIPS*, 2020.

[34] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015.

[35] C.-F. R. Chen, Q. Fan, and R. Panda, "Crossvit: Cross-attention multi-scale vision transformer for image classification," in *ICCV*, 2021.

[36] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *CVPR*, 2019.

[37] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W.-Y. Lo, J. Johnson, and G. Gkioxari, "Accelerating 3d deep learning with pytorch3d," *arXiv:2007.08501*, 2020.

[38] X. Jia, D. Blessing, X. Jiang, M. Reuss, A. Donat, R. Lioutikov, and G. Neumann, "Towards diverse behaviors: A benchmark for imitation learning with human demonstrations," in *ICLR*, 2024.

[39] J. Gu, F. Xiang, X. Li, Z. Ling, X. Liu, T. Mu, Y. Tang, S. Tao, X. Wei, Y. Yao, X. Yuan, P. Xie, Z. Huang, R. Chen, and H. Su, "Maniskill2: A unified benchmark for generalizable manipulation skills," in *ICLR*, 2023.