

Model-Based Diffusion Optimal Control for Multi-Robot Motion Planning

Zhilin He¹, Yorai Shaoul¹, Jiaoyang Li¹

¹Carnegie Mellon University
 {hectorh, yshaoul, jiaoyanl}@andrew.cmu.edu

Abstract—Multi-Robot Motion Planning in continuous environments, where robots must generate dynamically feasible, collision-free trajectories, is challenging due to the combinatorial growth of the joint trajectory space and the difficulty of enforcing dynamic feasibility and hard safety constraints. Recent approaches recast trajectory planning as probabilistic inference, sampling from a posterior over trajectories using diffusion models whose score functions are learned from demonstration data. While showing promising performance, these approaches are limited: they often rely on sizable demonstration datasets and struggle to rigorously enforce dynamics and hard safety constraints during sampling. To this end, we introduce Model-Based Diffusion Optimal Control (MDOC), a model-based diffusion planner that efficiently produces dynamically feasible trajectories without relying on data. Crucially, we show that MDOC’s safety mechanism—combining known dynamics models with Control Barrier Function-constrained projections—naturally scales to multi-robot planning settings through Conflict-Based Search. Across simulation experiments, this integrated method consistently outperforms representative baseline planners in sample efficiency, geometric smoothness, and success rate, while reducing computation time and producing collision-free trajectories. Code: <https://github.com/hhhhz1/mdoc>

I. INTRODUCTION

Consider a warehouse environment with numerous robots, where each robot attempts to reach a specified region while being subject to kinematic and dynamic constraints, such as acceleration limits and turning radius, and must avoid collisions with obstacles and other robots. The challenge, lies in efficiently coordinating movements to avoid collisions while navigating long, continuous trajectories under the combinatorial growth of the joint trajectory space in different environments.

Given the complexity of the full Multi-Robot Motion Planning (MRMP) problem, researchers have often resorted to solving simpler problems that approximate MRMP. One reformulation of the MRMP problem is known as Multi-Agent Path-Finding (MAPF) [43], where robots move in a discretized space (often a regular grid) and discrete time [33, 34]. In particular, the Conflict-Based Search (CBS) [39] family of algorithms [4, 6, 29] has been influential due to its strong scalability and safety guarantees. However, this MAPF-based abstraction relies on discretization assumptions, such as constant velocities, rectilinear and grid-based motions, which limit their applicability to continuous MRMP settings with dynamic feasibility and robot interactions.

More recent work has explored motion planning through probabilistic inference, sampling trajectory posteriors with

diffusion planners [7]. These planners leverage the strong generative capacity of diffusion models to learn a score function from expert demonstrations and represent complex, multimodal trajectory distributions in continuous spaces [17, 51]. By stochastically denoising trajectory priors, diffusion planners can produce smoother trajectories and avoid the restrictive assumptions imposed by grid-based simplifications. Building on these single-robot advances, recent work extends diffusion planners to MRMP [11, 30] and integrates them as low-level continuous-space planners within CBS for multi-robot settings [38].

However, existing diffusion planners remain fundamentally model-free, requiring large expert datasets to learn the score function and struggling to rigorously enforce dynamics during denoising the expert trajectories. As a result, they often rely on extensive learning from demonstrations and neglect existing model information, yielding trajectories that are not genuinely dynamics-aware. Additionally, current diffusion planners handle inter-robot collisions through soft constraints, where the hard safety constraints cannot be guaranteed, or through post-hoc safety projections [37, 48, 50, 10], which massively increase computational planning cost on full trajectory, making planners fail in large-scale multi-robot congested settings.

We propose Model-Based Diffusion Optimal Control (MDOC), a training-free diffusion planner that enforces collision avoidance and scales to MRMP without any demonstration data. Building on Model-Based Diffusion (MBD) [35], MDOC analytically estimates diffusion scores via Monte Carlo score ascent under a known dynamics model, interpreting denoising as stochastic optimal control so avoiding demonstration learning. Crucially, MDOC extends MBD to enforce rigorous collision avoidance through Control Barrier Function (CBF)-constrained projections directly inside the model-based diffusion rollouts. Capitalizing on these properties, we further propose MDOC-CBS, a scalable MRMP algorithm that resolves inter-robot collisions through the same safety mechanism while retaining the efficiency of high-level CBS. The contributions of this work are threefold:

- We develop MDOC, a model-based diffusion planner that enforces safety during sampling via CBF-constrained projections inside model-based rollouts, yielding dynamically feasible and collision-free trajectories.
- We propose MDOC-CBS, a CBS-based coordination that scales to 20 robots (40 in larger maps) while resolving inter-robot collisions via the same CBF-constrained pro-

jection by MDOC as the low-level planner.

- We evaluate our methods across diverse environments, start-goal configurations, constraint settings, and horizons and show that it improves success rate and trajectory quality while reducing computation and maintaining collision-free execution compared to representative baselines.

II. RELATED WORK

We review prior work on single-robot motion planning and multi-robot motion planning.

A. Single-Robot Motion Planning

Arguably, the most popular paradigm for single-robot motion planning is sampling. Sampling-based motion planning has been extensively studied through methods such as probabilistic roadmaps (PRM) [21, 22, 26] and rapidly exploring random trees (RRT) [27, 25, 23]. Although they provide probabilistic completeness [18, 42] and finite sampling guarantees [12], those methods struggle as problem geometry becomes more constrained and dynamics become more involved.

Trajectory optimization, mostly from control theory, formulates nonlinear robot motion planning problems directly over trajectories with constraints and dynamics. Sequential Linear Quadratic methods [41, 13] have been successfully used for legged robots [20, 14], while other works directly embed kinematic and dynamic limits for trajectory optimization [8, 31]. These methods can generate dynamically feasible trajectories, but solving large nonlinear programs becomes computationally demanding in cluttered environments.

Diffusion-based methods mitigate expensive optimization by learning a trajectory distribution and sampling candidates via Denoising Diffusion Probabilistic Models (DDPMs) [15, 17, 9]. Following these developments, Motion Planning Diffusion (MPD) [7] proposes learning and sampling diffusion models as prior and posterior trajectories for robot motion planning. Despite their efficiency, MPD still requires learning from large collections of expert trajectories. More recent methods combine diffusion models with control on Model Predictive Control [53], Model-Based Diffusion (MBD) [35] for trajectory optimization, and diffusion-based constraints [37]. However, safety mechanisms in those methods are either underexplored or enforced via inefficient projections.

B. Multi-Robot Motion Planning

Early works address the PSPACE-hardness of MRMP [16] by extending single-robot sampling-based planners into the composite configuration space, where each sample lies in the high-dimensional joint configuration space of all robots. Although those methods provide asymptotic optimality [40], they struggle in multi-robot settings because their search space grows exponentially with the number of robots.

To improve scalability, recent works utilized the Conflict-Based Search (CBS) framework [39] and decoupled the multi-robot planning problem into low-level single-robot planning and high-level conflict resolution. For example, KCBS [24] and db-CBS [33] design sampling-based low-level planners

with kinodynamic constraints, aiming for dynamic feasibility but still suffering from sampling inefficiency of kinodynamic, sampling-based tree expansion.

Multi-Robot Motion Planning with Diffusion Models (MMD) [38] combines MPD with CBS to provide collision-free paths in scalable multi-robot environments. Model-based approaches aim for trajectory optimization within three-dimensional holonomic cases [52] and annealing legs [49] without learning, but these methods still optimize all robots with a joint cost function, which limits scalability and makes hard safety constraints difficult to enforce.

Summary. We are the first to propose CBF-constrained projection with model-based diffusion as a single-robot planner, yielding dynamically feasible trajectories and ensuring safety during diffusion sampling, and to scale it to multi-robot settings through CBS.

III. PRELIMINARIES

Let us turn to formally defining the problem at hand and cover the relevant background. In what follows, we use a lower right index i to denote the robot identity, an upper right index h for the time step within the planning horizon, and an upper left index k for the diffusion step (e.g., ${}^k\tau_i^h$).

A. Problem Statement

In Multi-Robot Motion Planning (MRMP), we aim to convey a team of N robots from their start states to goal states while respecting each robot's dynamics and avoiding collisions with obstacles and each other. Let s_i encode the state of robot i , such as configuration position and velocity, and u_i encodes its control input. A trajectory can be represented as a sequence of state and control input pairs over a finite time horizon H as $\tau_i = [(s_i^1, u_i^1), (s_i^2, u_i^2), \dots, (s_i^H, u_i^H)]$. We denote by $\mathbf{s}^h := \{s_i^h\}_{i=1}^N$ and $\mathbf{u}^h := \{u_i^h\}_{i=1}^N$ the joint state and control of all robots at time h . Let $\boldsymbol{\tau}^{1:H} := \{\tau_i^{1:H}\}_{i=1}^N$ denote a multi-robot joint trajectory over time horizon 1 to H . We define the problem objective as

$$\mathcal{J}(\boldsymbol{\tau}^{1:H}) := \sum_{i=1}^N \left(\gamma_i \sum_{h=1}^H c(\tau_i^h) \right) + \sum_{i=1}^N \beta_i c_f(s_i^H), \quad (1)$$

where $\mathcal{J}$ indicates the objective function, c and c_f denote the intermediate and terminal cost functions (e.g., distance traveled between consecutive states and terminal distance to the goal), and γ_i and β_i are their associated positive weighting coefficients. Then, the MRMP problem is

$$\begin{aligned} \boldsymbol{\tau}^* &\in \arg \min_{\boldsymbol{\tau}^{1:H}} \mathcal{J}(\boldsymbol{\tau}^{1:H}) \\ \text{s.t. } s_i^{h+1} &= f(s_i^h, u_i^h), \quad \forall i, h = 1, \dots, H-1, \\ g^h(\mathbf{s}^h, \mathbf{u}^h) &\leq 0, \quad h = 1, \dots, H, \end{aligned} \quad (2)$$

with $s_i^{h+1} = f(s_i^h, u_i^h)$ representing the discrete-time dynamics, and $g^h(\mathbf{s}^h, \mathbf{u}^h) \leq 0$ generically collecting all obstacle-avoidance and inter-robot collision-avoidance constraints at time h to ensure safety.

B. Model-Based Diffusion for Trajectory Sampling

The trajectory for robot i , $\tau_i^{1:H}$, can be treated as a random variable and sampled by iterative denoising from Gaussian noise. Since this subsection focuses on single-robot motion planning, we omit the robot index i for brevity from now on.

We follow the standard DDPMs and denote by ${}^k\tau^{1:H}$ the noisy trajectory at diffusion step k :

$${}^{k+1}\tau^{1:H} = \sqrt{\bar{\alpha}_k} {}^1\tau^{1:H} + \sqrt{1 - \bar{\alpha}_k} \epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}), \quad (3)$$

where $\alpha_k := 1 - \beta_k$ controls the amount of signal preserved at diffusion step k under the noise-schedule coefficient β_k , $\bar{\alpha}_k := \prod_{n=1}^k \alpha_n$, and K is the total number of diffusion steps. The noise schedule is chosen such that the final noisy trajectory ${}^K\tau^{1:H}$ is close to standard Gaussian noise and can be initialized from $\mathcal{N}(0, \mathbf{I})$. Let ${}^1p({}^1\tau^{1:H})$ denote the target distribution over clean trajectories. We define the noisy marginals using the forward noising kernel q :

$${}^k p({}^k\tau^{1:H}) := \int q({}^k\tau^{1:H} \mid {}^1\tau^{1:H}) {}^1 p({}^1\tau^{1:H}) d{}^1\tau^{1:H}. \quad (4)$$

Based on this noisy marginal, a standard result in score-based diffusion is that the optimal reverse-time drift depends on the score function, $\mathbf{S}_k({}^k\tau^{1:H}) := \nabla_{{}^k\tau^{1:H}} \log {}^k p({}^k\tau^{1:H})$, where $\nabla_{{}^k\tau^{1:H}}$ denotes the gradient with respect to ${}^k\tau^{1:H}$. The score guides denoising toward higher-density regions of the noisy marginal ${}^k p$ at diffusion step k while removing noise. Consequently, the reverse update can be written in the score-guided form

$${}^{k-1}\tau^{1:H} = \frac{1}{\sqrt{\alpha_k}} \left({}^k\tau^{1:H} + (1 - \bar{\alpha}_k) \hat{\mathbf{S}}_k \right), \quad (5)$$

where $\hat{\mathbf{S}}_k \approx \mathbf{S}_k$ is a score estimate at step k . In standard DDPMs, $\hat{\mathbf{S}}_k$ is obtained from a learned noise predictor $\epsilon_\theta({}^k\tau^{1:H}, k)$, yielding the approximation

$$\mathbf{S}_k({}^k\tau^{1:H}) \approx -\frac{1}{\sqrt{1 - \bar{\alpha}_k}} \epsilon_\theta({}^k\tau^{1:H}, k). \quad (6)$$

Learning ϵ_θ (or a score network) typically requires a large amount of expert trajectories. When a dynamic model is available, MBD [35] avoids learning ϵ_θ by sampling control sequences ${}^k\bar{u}^{1:H-1}$ and using the known dynamics to roll them out into state trajectories ${}^k\tau^{1:H}$, then analytically constructing $\hat{\mathbf{S}}_k$ with denoised rollouts. Concretely, at reverse step k , we first sample denoised candidates from the Gaussian proposal induced by the forward kernel

$$\tilde{\tau}^{1:H} \sim \Omega_k := \mathcal{N}\left({}^k\tau^{1:H}, \left(\frac{1}{\bar{\alpha}_k} - 1\right) \mathbf{I}\right). \quad (7)$$

Let $\{\tilde{\tau}_m^{1:H}\}_{m=1}^M$ be M samples from Ω_k . We then evaluate each candidate under a target density ${}^1p(\cdot)$ and form the importance-weighted average

$${}^k\bar{\tau}^{1:H} = \frac{\sum_{m=1}^M {}^1p(\tilde{\tau}_m^{1:H}) \tilde{\tau}_m^{1:H}}{\sum_{m=1}^M {}^1p(\tilde{\tau}_m^{1:H})}, \quad (8)$$

which yields the Monte Carlo Score-Ascent (MCSA) estimator

$$\hat{\mathbf{S}}_k \approx -\frac{{}^k\tau^{1:H}}{1 - \bar{\alpha}_k} + \frac{\sqrt{\bar{\alpha}_k}}{1 - \bar{\alpha}_k} {}^k\bar{\tau}^{1:H}. \quad (9)$$

Together with Eqs (5)–(9), this yields a training-free diffusion sampler that iteratively (1) proposes candidates, (2) reweights them by ${}^1p({}^k\tau^{1:H})$, and (3) updates ${}^k\tau^{1:H}$ toward high-density trajectories. The remaining question is how to instantiate ${}^1p({}^k\tau^{1:H})$ so that high-probability samples correspond to low-cost, dynamically feasible, and collision-free trajectories. This can be answered through a planning-as-inference target distribution.

C. Planning as Inference

In the following, we omit both robot index i and diffusion step k for brevity. That is, we write the trajectory for robot i at diffusion step k , ${}^k\tau_i^{1:H}$, as $\tau^{1:H}$.

A standard relaxation of deterministic optimal control is to model soft optimality through a Boltzmann distribution, extensively studied in [44, 28, 36, 46, 45]:

$$p_J(\tau^{1:H}) \propto \exp\left(-\frac{\mathcal{J}(\tau^{1:H})}{\lambda}\right), \quad (10)$$

where $\lambda > 0$ is a temperature parameter. As $\lambda \rightarrow 0$, p_J concentrates around minimizers of $\mathcal{J}$.

Dynamic feasibility and safety can be encoded by a feasibility prior $p_{\text{feas}}(\tau^{1:H})$, so that ${}^1p(\tau^{1:H}) \propto p_{\text{feas}}(\tau^{1:H}) p_J(\tau^{1:H})$, whose support is restricted to dynamically feasible and collision-free trajectories. Conceptually, a hard-feasibility prior can be written as an indicator product

$$p_{\text{feas}}(\tau^{1:H}) \propto p_d(\tau^{1:H}) p_g(\tau^{1:H}), \quad (11a)$$

$$p_d(\tau^{1:H}) := \prod_{h=1}^{H-1} \mathbf{1}(s^{h+1} = f(s^h, u^h)), \quad (11b)$$

$$p_g(\tau^{1:H}) := \prod_{h=1}^H \mathbf{1}(g^h(s^h, u^h) \leq 0). \quad (11c)$$

However, directly using hard indicators is brittle in constrained environments: most candidates receive zero weight (“dead samples”), providing no useful score guidance for MCSA [32]. This motivates handling feasibility explicitly rather than through hard rejection. In practice, it is desirable to work with candidates that lie in the feasible set $\mathcal{D} \cap \mathcal{C}$, where $\mathcal{D}$ denotes dynamic feasibility by rollouts and $\mathcal{C} := \{\tau^{1:H} \mid g^h(s^h, u^h) \leq 0, \forall h = 1, \dots, H\}$ denotes the safety set induced by the inequality constraints. When trajectories are feasible, the MCSA importance weights in Eq. (8) can be computed using only the optimality term, $w_m \propto p_J(\tilde{\tau}_m^{1:H})$, without assigning zero weights to infeasible samples.

IV. METHODOLOGY

We present our Model-Based Diffusion Optimal Control (MDOC) planner and its multi-robot extension, MDOC-CBS. Given a known dynamics model, diffusion rollouts maintain dynamic feasibility, yielding trajectories in $\mathcal{D}$. Our contribution is to enforce safety during sampling, then scales to

Algorithm 1 Model-Based Diffusion Optimal Control

```

1: Input: Single robot start state  $s^1$ , goal state  $s^H$ , dynamics  $f(\cdot)$ ,
   safety set  $\mathcal{C}$ , and finite time horizon  $H$ .
2: Output: Clean denoised trajectory  ${}^1\tau^{1:H}$ .
3: Sample an initial noisy trajectory  ${}^K\tau^{1:H} \sim \mathcal{N}(0, I)$ .
4: for  $k = K$  to 1 do
5:   for  $m = 1$  to  $M$  do
6:     Sample a candidate trajectory:
        $\tilde{\tau}_m^{1:H} \sim \mathcal{N}\left(\frac{{}^k\tau^{1:H}}{\sqrt{\alpha_k}}, \left(\frac{1}{\alpha_k} - 1\right) \mathbf{I}\right)$ .
7:     Recover a dynamically feasible and safe trajectory:
        $\tau_m^{1:H} \leftarrow \text{ROLLOUTWITHCBF}(\tilde{\tau}_m^{1:H}, s^1, s^H, f(\cdot), \mathcal{C})$ .
8:   end for
9:   Calculate Monte Carlo mean  $\bar{\tau}^{1:H}$  via Eq. (16).
10:  Estimate score  $\hat{\mathbf{S}}_k$  via Eq. (9).
11:  Compute Monte Carlo score ascent via (5):
        ${}^{k-1}\tau^{1:H} \leftarrow \frac{1}{\sqrt{\alpha_k}} \left( {}^k\tau^{1:H} + (1 - \alpha_k) \hat{\mathbf{S}}_k \right)$ .
12: end for
13: return  ${}^1\tau^{1:H}$ .

```

MRMP, by introducing a feasibility operator $\mathcal{F}$ that maps each candidate into $\mathcal{D} \cap \mathcal{C}$. Specifically, $\mathcal{F}$ applies CBF-constrained projections on controls within the model-based diffusion rollouts.

A. Model-Based Diffusion Optimal Control (MDOC)

MDOC is a single-robot motion planner, and thus we omit the robot index i for brevity in the following. MDOC follows the diffusion paradigm. At each diffusion step k , it invokes a sampler that proposes a nominal control sequence ${}^k\bar{u}^{1:H-1}$ (with ${}^k\bar{u}^h$ at time h), which may violate safety constraints. To maintain safety, MDOC projects ${}^k\bar{u}^{1:H-1}$ to a safe sequence ${}^k u^{1:H-1}$ that satisfies $\mathcal{C}$ via a feasibility operator $\mathcal{F}$, which enforces a discrete-time CBF condition during the model-based rollout.

Concretely, we require a barrier function $b({}^k s^h)$ (e.g., $b({}^k s^h) = \|{}^k s^h - s_{\text{obs}}\|^2 - r^2$, so $b({}^k s^h) \geq 0$ enforces a minimum separation r from the obstacle) that satisfies the discrete-time CBF condition $b({}^k s^{h+1}) \geq (1 - \gamma\Delta h) b({}^k s^h)$, where Δh is the time step size, meaning that the safety margin $b(\cdot)$ should not decay faster than rate γ . We locally linearize the dynamics and the barrier function around $({}^k s^h, {}^k \bar{u}^h)$:

$${}^k s^{h+1} \approx {}^k \hat{s}^{h+1} + {}^k B^h ({}^k u^h - {}^k \bar{u}^h), \quad {}^k \hat{s}^{h+1} := f({}^k s^h, {}^k \bar{u}^h), \quad (12)$$

where ${}^k B^h := \frac{\partial f(s, u)}{\partial u} \Big|_{({}^k s^h, {}^k \bar{u}^h)}$ is the Jacobian that locally maps control deviations to changes in the next state. Applying the same first-order approximation to $b(\cdot)$ yields the control-affine inequality [1, 2, 3]

$$\begin{aligned} \nabla b({}^k \hat{s}^{h+1})^\top {}^k B^h {}^k u^h &\geq \\ (1 - \gamma\Delta h) b({}^k s^h) - b({}^k \hat{s}^{h+1}) + \nabla b({}^k \hat{s}^{h+1})^\top {}^k B^h {}^k \bar{u}^h, \end{aligned} \quad (13)$$

which is equivalent to ${}^k a^{h\top} {}^k u^h \geq {}^k d^h$, where

$$\begin{aligned} {}^k a^{h\top} &:= \nabla b({}^k \hat{s}^{h+1})^\top {}^k B^h, \\ {}^k d^h &:= (1 - \gamma\Delta h) b({}^k s^h) - b({}^k \hat{s}^{h+1}) + {}^k a^{h\top} {}^k \bar{u}^h. \end{aligned} \quad (14)$$

Algorithm 2 MDOC-CBS

```

1: Input:  $N$  robots with their start states  $s^1$ , goal states  $s^H$ ,
   dynamics  $f(\cdot)$ , safety set  $\mathcal{C}_{\text{obs}}$  that encode obstacle-avoidance
   constraints, and finite time horizon  $H$ .
2: Output: Trajectories  $\tau^{1:H}$ .
3: Create CT root node  $V_{\text{root}}$ .
4:  $\text{CT} \leftarrow \{V_{\text{root}}\}$ .
5: for  $i = 1$  to  $N$  do
6:    $V_{\text{root}}. \mathcal{C}_i \leftarrow \mathcal{C}_{\text{obs}}$ .
7:    $V_{\text{root}}. \tau_i^{1:H} \leftarrow \text{MDOC}(s_i^1, s_i^H, f(\cdot), V_{\text{root}}. \mathcal{C}_i, H)$ .
8:   if  $V_{\text{root}}. \tau_i^{1:H}$  is not feasible then
9:     return “no solution”.
10:  end if
11: end for
12: while  $\text{CT} \neq \emptyset$  do
13:    $V \leftarrow \arg \min_{V' \in \text{CT}} \text{NUMCONFLICTS}(V'. \tau^{1:H})$ .
14:   Remove  $V$  from CT.
15:   if  $V. \tau^{1:H}$  is conflict-free then
16:     return  $V. \tau^{1:H}$ .
17:   end if
18:    $s_i^h, s_j^h, h, i, j \leftarrow \text{GETONECONFLICTPAIR}(V. \tau^{1:H})$ .
19:   for  $z \in \{i, j\}$  do
20:      $V_z \leftarrow V. \text{copy}()$ .
21:      $V_z. \mathcal{C}_z \leftarrow V_z. \mathcal{C}_z \cup \{\langle z, \text{Sphere}(s_{i+j-z}^h, h) \rangle\}$ .
22:      $V_z. \tau_z^{1:H} \leftarrow \text{MDOC}(s_z^1, s_z^H, f(\cdot), V_z. \mathcal{C}_z, H)$ .
23:     if  $V_z. \tau_z^{1:H}$  is feasible then
24:       Add  $V_z$  to CT.
25:     end if
26:   end for
27: end while
28: return “no solution”.

```

These inequalities define a safe-control set in the control space ${}^k \mathcal{K}^h := \{u \mid {}^k A^h u \geq {}^k D^h\}$, where ${}^k D^h$ is the vector collecting all scalar offsets ${}^k d^h$.

Stacking the half-spaces from all obstacles yields the polyhedral safe-control set ${}^k \mathcal{K}^h = \{u \mid {}^k A^h u \geq {}^k D^h\}$, where each row ${}^k a_j^{h\top}$ of ${}^k A^h$ and entry ${}^k d_j^h$ of ${}^k D^h$ correspond to one linearized CBF constraint. For speed inside diffusion sampling, we enforce each violated half-space via a closed-form projection. Given a single constraint ${}^k a^{h\top} {}^k u^h \geq {}^k d^h$, the projection for nominal control ${}^k \bar{u}^h$ is

$${}^k u^h \leftarrow \begin{cases} {}^k \bar{u}^h, & {}^k a^{h\top} {}^k \bar{u}^h \geq {}^k d^h, \\ {}^k \bar{u}^h + \frac{{}^k d^h - {}^k a^{h\top} {}^k \bar{u}^h}{\|{}^k a^h\|^2} {}^k a^h, & \text{otherwise.} \end{cases} \quad (15)$$

The correction term moves ${}^k \bar{u}^h$ along the constraint normal ${}^k a^h$ by the minimal amount so that ${}^k a^{h\top} {}^k u^h = {}^k d^h$, which is the Euclidean projection onto the corresponding half-space and solves $\min_{{}^k u^h} \frac{1}{2} \|{}^k u^h - {}^k \bar{u}^h\|^2$ subject to ${}^k a^{h\top} {}^k u^h \geq {}^k d^h$.

Therefore, given ${}^k \bar{u}^{1:H-1}$, we compute a feasible rollout by projecting per step ${}^k \bar{u}^h$ to ${}^k u^h$ through the half-space projection implied by Eq. (15). In MDOC, $\mathcal{F}$ is thus implemented by iterating over H : (1) projecting each proposed control ${}^k \bar{u}^h$ onto ${}^k \mathcal{K}^h$ to enforce safety and (2) rolling out the known dynamics one step to obtain ${}^k s^{h+1}$, so dynamic feasibility is not broken.

We now combine (1) the MCSA sampler from Section III-B

and (2) the feasibility operator $\mathcal{F}$. At each diffusion step k , MDOC samples M potentially unsafe candidates $\{\tilde{\tau}_m^{1:H}\}_{m=1}^M$ from the Gaussian proposal with Eq. (7) parameterized by controls ${}^k\tilde{u}^{1:H-1}$, then recovers dynamically feasible and CBF-constrained trajectories $\{\tau_m^{1:H}\}_{m=1}^M$ by rolling out ${}^k\tilde{u}_m^{1:H-1}$ through the known dynamics and applying the projection with Eq. (15), namely $\tau_m^{1:H} = \mathcal{F}(\tilde{\tau}_m^{1:H})$, computes Monte Carlo mean with

$${}^k\bar{\tau}^{1:H} = \frac{\sum_{m=1}^M {}^1p(\tau_m^{1:H}) \tau_m^{1:H}}{\sum_{m=1}^M {}^1p(\tau_m^{1:H})}, \quad (16)$$

and forms the score estimate from Eq. (9) to update the noisy trajectory through Eq. (5), as summarized in Algorithm 1. As a result, all trajectories used to compute the Monte Carlo estimate in Eq. (9) satisfy the dynamics and constraints.

B. MDOC-CBS

We now show how the mechanisms in MDOC can directly generalize to support planning for multi-robot settings via CBS [39]. Briefly, the CBS algorithm decomposes the MRMP problem into two levels: a low-level where each robot plans for itself, and a high-level, where the planned trajectories are surveyed, conflicts (i.e., inter-robot collisions) are identified, and constraints are imposed on colliding robots such that they will avoid the collision configurations upon replanning.

Specifically, MDOC-CBS, as shown in Algorithm 2, first creates a root node V_{root} and stores it in a priority queue termed the constraint tree (CT). The root node includes (1) an constraint set $V_{\text{root}}.\mathcal{C}_i$ for each robot i , initialized with obstacle-avoidance constraints, and (2) a joint trajectory $V_{\text{root}}.\tau^{1:H}$, constructed by planning for each robot with MDOC over the horizon H .

CBS iteratively selects the least cost node V from the CT (in our case, the one with the fewest conflicts) and evaluates it. If $V.\tau^{1:H}$ is conflict-free, it returns it as a solution. Otherwise, if a conflict is found between robots i and j at time h , CBS branches node V into two new CT nodes, V_i and V_j , for robots i and j : each child node copies the constraint sets and trajectories from V and adds a new constraint forbidding robot i (resp. j) from entering a small workspace sphere centered at robot j 's state s_j^h at time h , with a safe margin r . CBS then replans the trajectories of robots i and j using the low-level MDOC planner under the updated constraint sets $V_i.\mathcal{C}_i$ and $V_j.\mathcal{C}_j$, enforced by the single-robot feasibility operator $\mathcal{F}$ via CBF-constrained projections within the model-based diffusion rollouts, as illustrated in Algorithm 1. When replanning robot i , each CBS constraint generated from a conflict with robot j is represented as a time-indexed forbidden sphere centered at the conflicting state of robot j . We convert the pairwise CBF $b_{ij}(s_i, s_j)$ into a single-robot barrier by fixing the second argument to this sphere center, that $b_{ij}^h(s_i^h) = b_{ij}(s_i^h, s_j^h)$, where s_j^h is fixed by the CBS constraint. If $b_{ij}(s_i^h, s_j^h) = \|s_i^h - s_j^h\|^2 - r^2$, then $b_{ij}^h(s_i^h) = \|s_i^h - s_j^h\|^2 - r^2 \geq 0$, enforced by $\mathcal{F}$ with $b_{ij}^{h+1} \geq (1 - \gamma\Delta h)b_{ij}^h$ during diffusion rollouts. The two new CT nodes, with updated trajectories $V_i.\tau^{1:H}$ and

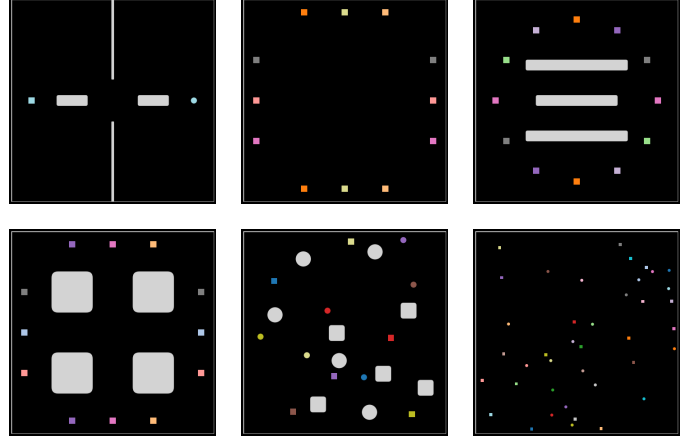


Fig. 1: Illustrations of Narrow map, Empty map, Conveyor map (Top Row), Drop-Region map, Random map, and Empty Large map (Below) used in MRMP experiments where gray objects represent obstacles. Colorful squares and spheres indicate the start and goal of robots. For the Empty and Conveyor maps, we consider a dual swap (Circle Setup and Weave Setup) between two robots positioned on squares of the same color.

$V_j.\tau^{1:H}$, are added to the CT priority queue. CBS then pops the node with the fewest conflicts for further expansion.

V. EXPERIMENTAL ANALYSIS

We aim to explore the performance of our method through various environments to address the following three questions:

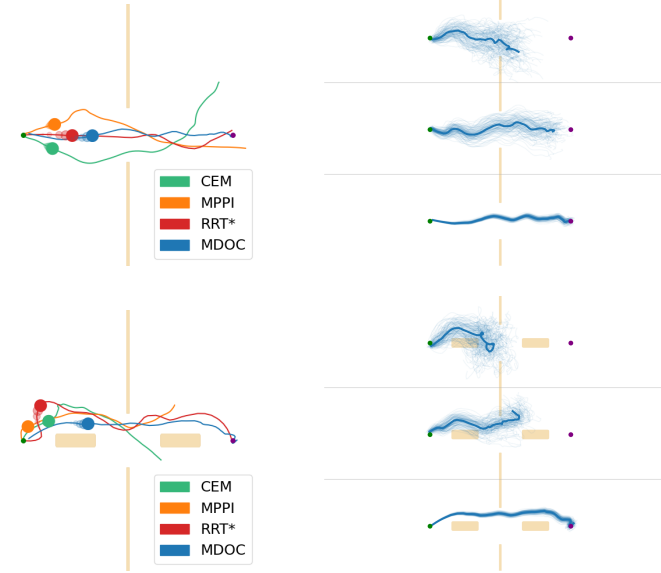
- What advantages does MDOC offer over state-of-the-art (SOTA) single-robot planners?
- How does MDOC-CBS compare to SOTA MRMP planners in terms of success rate, trajectory smoothness, and the ability to ensure constraints?
- Is MDOC-CBS robust and scalable with respect to map size, planning horizon, and number of robots?

Experimental Setup. We evaluate our method MDOC and its MRMP version MDOC-CBS on various map types (illustrated in Fig. 1, : (1) we use the Narrow map of size 2×2 to evaluate MDOC in single-robot settings; (2) we use the Empty map, Conveyor map [38], Drop-Region map [38], and Random Map (with an obstacle density of 13.7%), all of size 2×2 , to evaluate MDOC-CBS in multi-robot settings; (3) we use Empty Large maps of size 4×4 and 6×6 for multi-robot scalability evaluation. The robot radius is fixed at 0.05 for all maps.

We consider three start-goal pairs setups: (1) Circle Setup requires robots to symmetrically swap positions between opposite points on the perimeter; (2) Weave Setup requires robots to exchange positions along uniformly spaced boundary points; (3) Random Setup randomly generates start-goal pairs in maps. Setup demonstrations are shown

Baselines. We compare our method, MDOC, with SOTA trajectory optimization and planning methods: (1) CEM [5], a sampling-based trajectory optimizer using importance-weighted elite selection; (2) MPPI [47], a stochastic model-

Fig. 2: Results of MDOC and baseline methods on Narrow map (top row) and More-Constrained Narrow map (bottom row). The left panels plot, for each planner, the collision-free trajectory with the highest success probability. Right panels depict MDOC rollouts at 90%, 50%, and 10% of the diffusion denoising process, showing how the distribution converges toward successful passages.



	CEM	MPPI	RRT*	MDOC
Top	2.1 ± 0.2	3.2 ± 0.3	66.1 ± 6.0	100.0
Bottom	0.2 ± 0.1	0.4 ± 0.1	41.7 ± 3.4	100.0

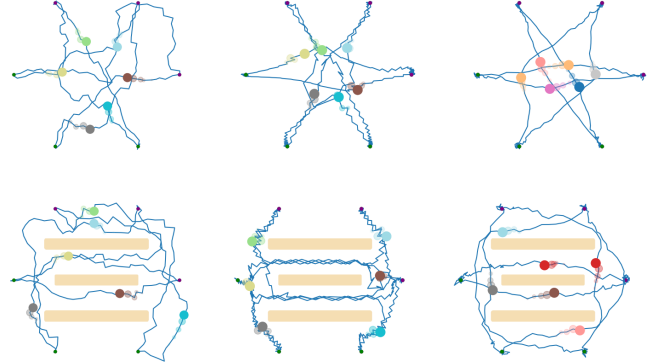
TABLE I: “Pass&Free”-Yield (%) ($\uparrow$) of MDOC and baseline methods, which is defined as the percentage of candidate trajectories that both traverse the bottleneck (Pass) and remain collision-free (Free) under a fixed proposal sample size $M = 128$ per planning step.

predictive control method based on path integral sampling; and (3) RRT* [19], a sampling-based asymptotically optimal motion planner. We also compare **MDOC-CBS** with SOTA CBS-based MRMP methods: (4) CBS [39] and its variant Enhanced CBS (ECBS) [4], multi-agent path finding planners with grid-based A* in the discretized configuration space as its low-level planner; (5) KCBS [24], a continuous-space variant of CBS with kinematic RRT* as its lower-level planner; (6) MMD-CBS [38], a continuous-space variant of CBS with model-free diffusion, trained on RRT* trajectories, as its low-level planner. Note that we mainly focus on CBS-based MRMP planners, since MMD-CBS outperforms “composite” methods, which directly optimize joint constraints and objectives without CBS [38].

A. Experimental Results

We conduct experiments on an NVIDIA RTX A4500 GPU with 20 GB of memory and 12 CPU cores. We run 10 trials

Fig. 3: Results on Empty map (Top Row) and Conveyor map (Below) with 6 robots on Circle setup. Columns display 3 algorithms: KCBS (Left), MMD-CBS (Mid), and our MDOC-CBS (Right). We show trajectories without post-smoothness.



Map	Method	$\mathbf{P} \downarrow$	$\mathbf{G} (1e-4) \downarrow$
Empty	KCBS	2.2 ± 0.07	20.5 ± 2.75
	MMD-CBS	2.9 ± 0.06	60.6 ± 5.33
	MDOC-CBS	1.8 ± 0.03	6.7 ± 0.05
Conveyor	KCBS	2.7 ± 0.10	20.7 ± 2.12
	MMD-CBS	3.9 ± 0.11	85.0 ± 5.45
	MDOC-CBS	2.8 ± 0.04	12.9 ± 1.40

TABLE II: Comparison of methods in terms of average path length $\mathbf{P}$ and geometric (Laplacian) smoothness $\mathbf{G}$ over success trails of all agent numbers, evaluated on the Empty map and Conveyor map in the Circle setup.

per method and map, each with a different random seed, under identical map conditions. We use a single-integrator model in the main experiments for clarity.

Why Model-Based Diffusion in Motion Planning? As illustrated in the right panel of Fig. 2, we showcase the sample distribution at 90%, 50%, and 10% of the diffusion denoising process. We observe that the early stage (90%) distribution is wide, covering multiple homotopy classes, and exploring diverse passages. As iterations progress (50%), the distribution is denoising towards feasible and low-cost paths by score guidance augmented with dynamics and constraint projections, traversing the bottleneck in both maps and safely avoiding obstacles in More-Constrained Narrow (bottom row), and finally converges to a high-quality feasible trajectory at late stage (10%). CEM or MPPI refine the control sequence through importance weighting or elite resampling, but mainly rely on repeated stochastic resampling, which often leads to getting stuck in wrong homotopy classes or failing in highly multimodal or constrained environments. MDOC establishes a well-conditioned gradient structure over the trajectory distribution and explicitly incorporates dynamics and constraints into the denoising process, so that every reverse step is a model-informed guidance rather than a blind perturbation. Together, these properties turn motion planning into a model-based

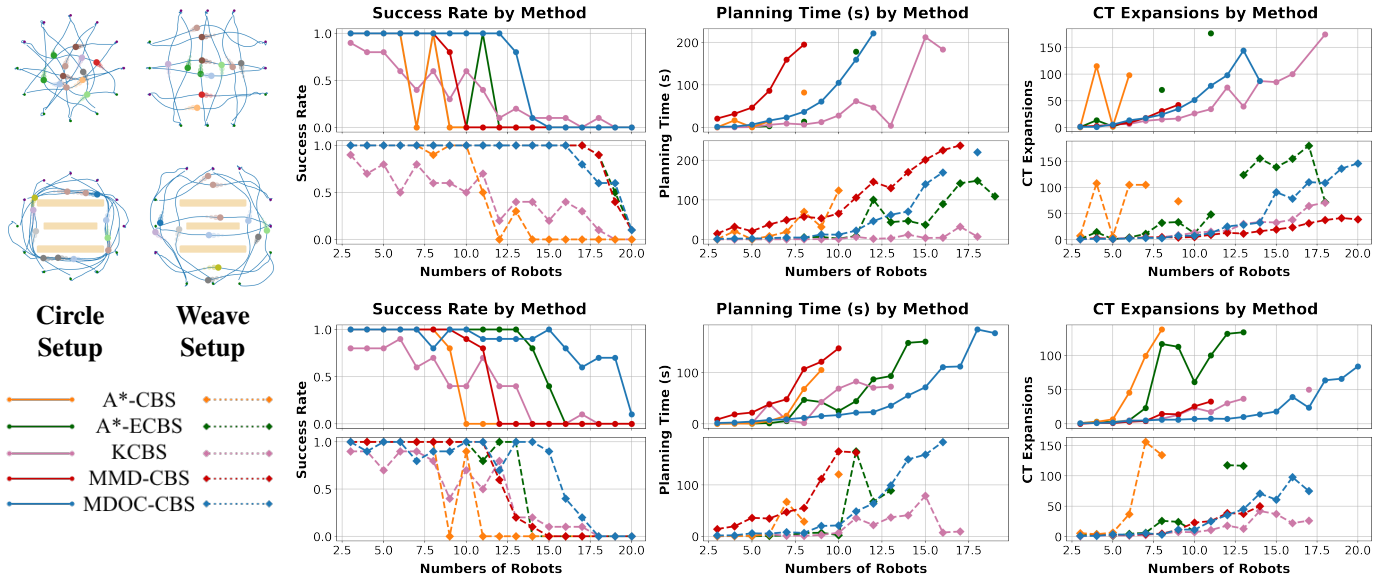


Fig. 4: Performance of MRMP methods on Empty map (top row) and Conveyor map (bottom row) with a 500-second time limit. For each subfigure, the top plot shows the Circle Setup (circular markers with solid lines), and the bottom plot shows the Weave Setup (diamond markers with dashed lines). The planning time and CT expansions are averaged on successful trials, and outliers are included in the averages but omitted beyond the 95th percentile for better visualization, especially in CT expansions.

diffusion process, where the dynamics model and constraints shape the score instead of relying solely on sampled rollouts. This allows each candidate trajectory to become progressively more feasible and cost-efficient within a single multi-step refinement procedure, achieving higher effectiveness under the same sampling budget compared to other optimizers that depend purely on random perturbation resampling. “Pass&Free”-Yield (PF-Yield) from Table I further verifies that MDOC achieves a 100.0% effective sample rate, while CEM and MPPI remain significantly lower on both narrow maps.

As a tree-search-based geometric planner, RRT* explores reachable paths by randomly sampling states and incrementally expanding a search tree, but it does not directly optimize trajectory cost or enforce dynamic feasibility. To quantify its effective sampling efficiency in narrow passages, we compute PF-Yield for RRT* as the fraction of expanded nodes whose associated paths both traverse the bottleneck and remain collision-free (Pass & Free) over all expanded nodes. Comparing PF-Yield across RRT* and MDOC therefore places a geometric planner and a trajectory optimizer on the same footing, by asking: out of all samples or rollouts under a fixed budget, how many actually produce useful, bottleneck-traversing, collision-free candidates? Under this metric, RRT* attains PF-Yield values of 66.1% and 41.7% on the two maps, substantially lower than MDOC’s 100%. This indicates that, under a limited sample budget, while RRT* possesses global reachability and probabilistic completeness, it still wastes many samples in unproductive homotopy classes and struggles to efficiently converge to a feasible and smooth solution in bottlenecked environments. In contrast, MDOC, through diffusion

denoising in trajectory space, achieves continuous convergence from global exploration to local refinement guided by dynamics and constraints, thereby completing path selection and trajectory optimization within a single model-based diffusion process. Therefore, we consider MDOC as a model-based diffusion planner that inherits the global exploration behavior of RRT* while also enjoying the optimization capabilities of MPPI and CEM, using the underlying dynamics model and hard constraints to steer the diffusion updates, thereby achieving higher feasibility and convergence efficiency.

Performance for MRMP on 2×2 Maps. We first evaluate the scalability of MDOC-CBS on the Empty map and Conveyor map with the Circle and Weave setups up to 20 robots (Fig. 4). The experiment uses fixed horizon $H = 64$. Overall, the results show that MDOC-CBS (blue lines) outperforms baseline methods under the same map and start-goal conditions in most cases. In contrast, MMD-CBS fails beyond 10 robots in the Empty Circle setup and 12 in the Conveyor Circle setup, while KCBS’s success rate steadily degrades after 5 robots. At the same time, MDOC-CBS achieves $5 \times -10 \times$ lower planning time than MMD-CBS at comparable CT expansion levels, since it computes the trajectory score analytically through MCSA with model-based rollouts rather than repeatedly invoking a large diffusion network.

Moreover, our MDOC-CBS consistently produces shorter and smoother trajectories than KCBS and MMD-CBS (Table II). This improvement stems from the model-based diffusion objective with MCSA, which drives rich exploration directly in continuous state space while regularizing dynamic coherence. KCBS, which relies on discrete RRT* primitives,

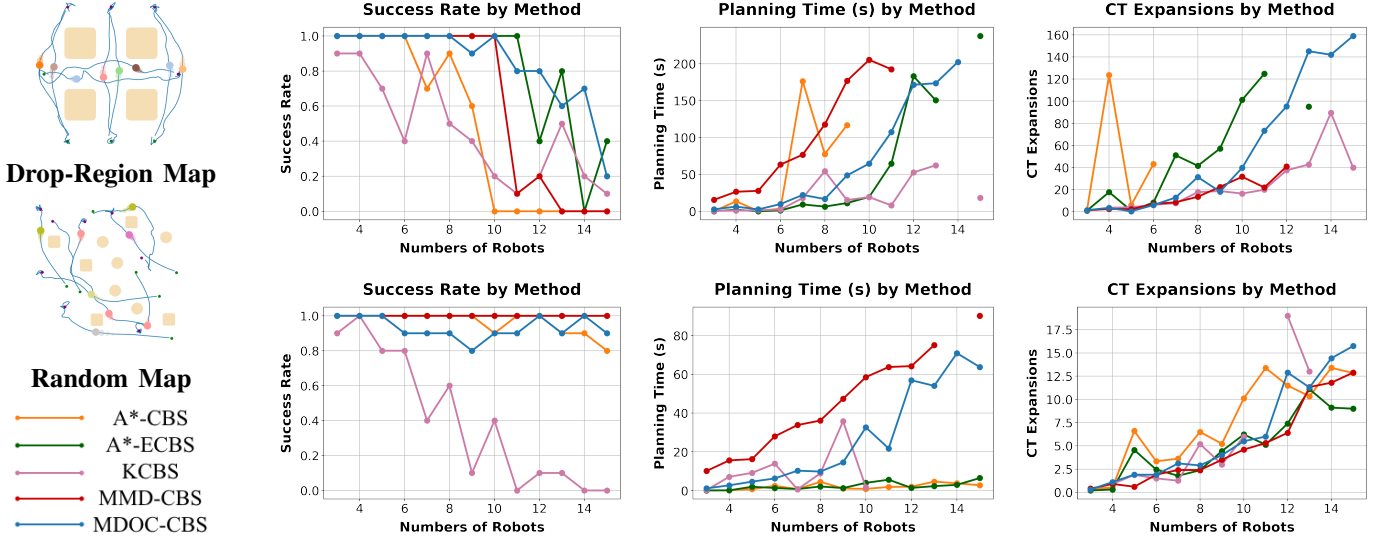


Fig. 5: Performance of MRMP methods on Drop-Region map with the Weave Setup (top row) and Random map with the Random Setup (bottom row) with a 500-second time limit. The planning time and CT expansions are averaged on successful trials, and outliers are included in the averages but omitted beyond the 95th percentile for better visualization, especially in CT expansions.

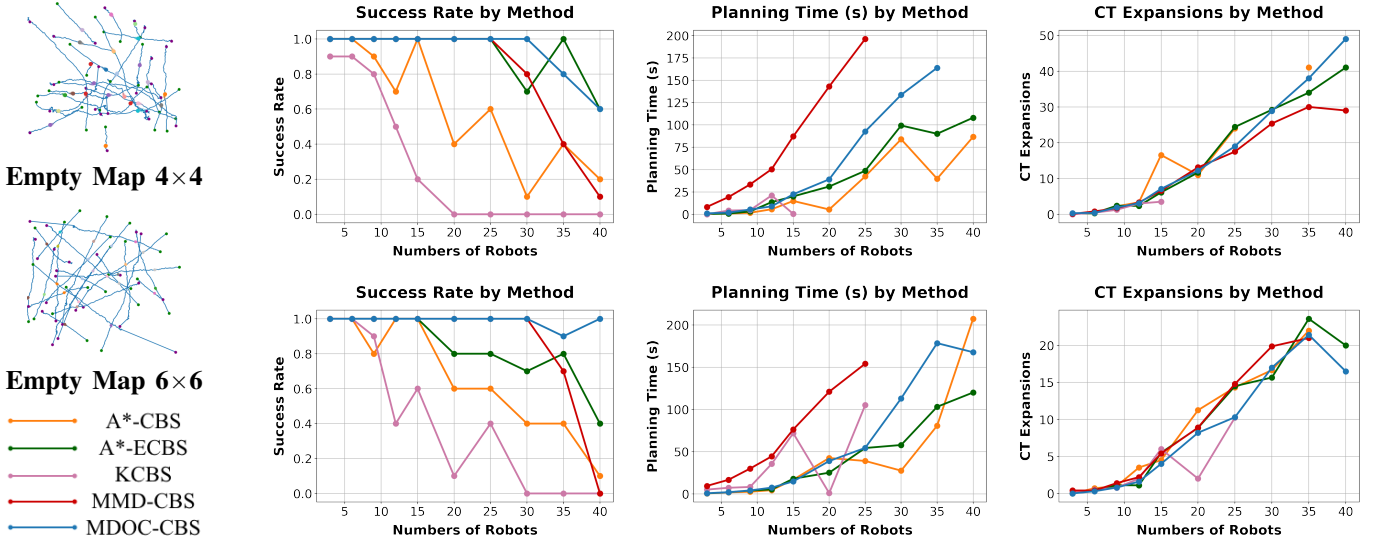


Fig. 6: Performance of MRMP methods on the larger 4×4 (top row) and 6×6 (bottom row) Empty maps with a 500-second time limit, both with the Random Setup. The planning time and CT expansions are averaged on successful trials, and outliers are included in the averages but omitted beyond the 95th percentile for better visualization, especially in CT expansions.

tends to generate irregular, jittery motions, and MMD-CBS inherits these artifacts because its denoising objective is trained to imitate RRT*-generated demonstrations. In contrast, MDOC-CBS actively exploits open free space and promotes dynamically coordinated motion across agents (Fig. 3), yielding trajectories that are globally more symmetric and better synchronized. The difference is most pronounced in the Conveyor map, as shown in the bottom row of Fig. 3. MMD-CBS inherits the demonstration’s “queue-through-the-corridor” behavior: all 6 robots enter the narrow passage,

causing oscillation and congestion at the bottleneck, which inflates path length and degrades geometric smoothness. In contrast, MDOC-CBS does not force all robots into the same homotopy. Through model-based score ascent and continuous-space exploration, only 2 robots choose the corridor, while the remaining robots naturally distribute themselves around the outer free space. This leads to significantly shorter average path lengths, fewer high-level CT expansions, and substantially smoother continuous trajectories while ensuring a safety margin. These gains arise from combining diffusion-based

exploration with a model-based low-level objective within CBS, thereby improving scalability in dense settings.

We further evaluate MDOC-CBS on a Drop-Region map with four tightly packed central obstacles and a Random map with uniformly sampled start-goal pairs (Fig. 5). In both settings, MDOC-CBS and MMD-CBS achieve similar success rates and clearly outperform KCBS, while MDOC-CBS consistently attains lower planning time at comparable CT expansion levels, indicating robustness across different map structures. However, MDOC-CBS exhibits higher variability: its success rate does not always reach 100% in random maps, and most failures occur when a robot collides with an obstacle and no valid rollout is returned, reflecting higher Monte Carlo score ascent variance in compressed free space with tight random constraints. This suggests a nontrivial trade-off between open-space exploration and aggressive safety tightening within the CBF-constrained projection during model-based diffusion rollouts. Even so, MDOC-CBS solves instances with up to 15 robots, and we expect stability to improve with better variance control and constraint projections in future work.

Longer-Horizon MRMP on Larger Maps. We present a robustness study of MDOC-CBS on larger maps with longer horizons ($H = 128$ for larger 4×4 and $H = 196$ for larger 6×6). As shown in the Fig. 6, MDOC-CBS remains robust as the problem size grows, achieving the highest success rate up to 40 robots. In contrast, MMD-CBS degrades sharply beyond 30 robots: increasing H directly lengthens the denoising chain and inflates the inference cost of its model-free single-robot diffusion model, so many runs hit the time limit before CBS can complete the high-level search. MDOC-CBS, by comparison, scales much more gracefully with horizon length. Because MDOC performs model-based diffusion rollouts in parallel under known dynamics, the per-iteration cost grows sub-linearly with H compared to model-free denoising, and even when CBS introduces additional constraints on larger maps, the computation can be amortized across robots and time steps. This makes MDOC-CBS substantially more robust for long-horizon planning and explains its superior performance on the 6×6 map.

VI. CONCLUSION

In this work, we propose MDOC, a CBF-constrained model-based diffusion planner that satisfies dynamical feasibility and safety constraints simultaneously. We also extend the planner to the multi-robot settings to solve MRMP. Results showcase the superior performance of our method in various environmental setups. Future directions include optimizing Monte Carlo variance and constraint projection efficiency, adapting the planner to online tasks with receding-horizon methods, and exploring richer dynamics models and multi-robot variants such as Prioritized Planning.

ACKNOWLEDGMENTS

This work was partially supported by the National Science Foundation under Grants #2328671 and #2441629.

REFERENCES

- [1] Ayush Agrawal and Koushil Sreenath. Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation. In *Robotics: Science and Systems*, pages 1–10, 2017.
- [2] Aaron D Ames, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs with application to adaptive cruise control. In *Conference on Decision and Control*, pages 6271–6278, 2014.
- [3] Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Genaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *European Control Conference*, pages 3420–3431, 2019.
- [4] Max Barer, Guni Sharon, Roni Stern, and Ariel Felner. Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem. In *International Symposium on Combinatorial Search*, pages 19–27, 2014.
- [5] Zdravko I Botev, Dirk P Kroese, Reuven Y Rubinstein, and Pierre L’ecuyer. The cross-entropy method for optimization. In *Handbook of statistics*, volume 31, pages 35–59. Elsevier, 2013.
- [6] Eli Boyarski, Ariel Felner, Roni Stern, Guni Sharon, Oded Betzalel, David Tolpin, and Eyal Shimony. ICBS: The improved conflict-based search algorithm for multi-agent pathfinding. In *International Symposium on Combinatorial Search*, pages 223–225, 2015.
- [7] Joao Carvalho, An T Le, Mark Baierl, Dorothea Koert, and Jan Peters. Motion planning diffusion: Learning and planning of robot motions with diffusion models. In *International Conference on Intelligent Robots and Systems*, pages 1916–1923, 2023.
- [8] Yufan Chen, Mark Cutler, and Jonathan P How. Decoupled multiagent path planning via incremental sequential convex programming. In *International Conference on Robotics and Automation*, pages 5954–5961, 2015.
- [9] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.
- [10] Jacob K Christopher, Stephen Baek, and Ferdinando Fioretto. Constrained synthesis with projected diffusion models. In *Advances in Neural Information Processing Systems*, pages 89307–89333, 2024.
- [11] Apan Dastider, Hao Fang, and Mingjie Lin. APEX: Ambidextrous dual-arm robotic manipulation using collision-free generative diffusion models. In *International Conference on Intelligent Robots and Systems*, pages 9526–9533, 2024.
- [12] Dror Dayan, Kiril Solovey, Marco Pavone, and Dan Halperin. Near-optimal multi-robot motion planning with finite sampling. *IEEE Transactions on Robotics*, 39(5): 3422–3436, 2023.
- [13] Farbod Farshidian, Edo Jelavic, Asutosh Satapathy,

- Markus Gifftthaler, and Jonas Buchli. Real-time motion planning of legged robots: A model predictive control approach. In *International Conference on Humanoid Robotics*, pages 577–584, 2017.
- [14] Ruben Grandia, Fabian Jenelten, Shaohui Yang, Farbod Farshidian, and Marco Hutter. Perceptive locomotion through nonlinear model-predictive control. *IEEE Transactions on Robotics*, 39(5):3402–3421, 2023.
- [15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, pages 6840–6851, 2020.
- [16] John E Hopcroft and Gordon T. Wilfong. Reducing multiple object motion planning to graph searching. *SIAM Journal on Computing*, 15(3):768–785, 1986.
- [17] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.
- [18] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *International Journal of Robotics Research*, 30(7):846–894, 2011.
- [19] Sertac Karaman, Matthew R Walter, Alejandro Perez, Emilio Frazzoli, and Seth Teller. Anytime motion planning using the RRT. In *International Conference on Robotics and Automation*, pages 1478–1483, 2011.
- [20] Sotaro Katayama, Masaki Murooka, and Yuichi Tazaki. Model predictive control of legged and humanoid robots: Models and algorithms. *Advanced Robotics*, 37(5):298–315, 2023.
- [21] Lydia E Kavraki, Petr Svestka, J-C Latombe, and Mark H Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, 1996.
- [22] Lydia E Kavraki, Mihail N Kolountzakis, and J-C Latombe. Analysis of probabilistic roadmaps for path planning. *IEEE Transactions on Robotics and automation*, 14(1):166–171, 1998.
- [23] Michal Kleinbort, Kiril Solovey, Zakary Littlefield, Kostas E Bekris, and Dan Halperin. Probabilistic completeness of RRT for geometric and kinodynamic planning with forward propagation. *IEEE Robotics and Automation Letters*, 4(2):277–283, 2019.
- [24] Justin Kottinger, Shaul Almagor, and Morteza Lahijanian. Conflict-based search for multi-robot motion planning with kinodynamic constraints. In *International Conference on Intelligent Robots and Systems*, pages 13494–13499, 2022.
- [25] James J Kuffner and Steven M LaValle. RRT-connect: An efficient approach to single-query path planning. In *International Conference on Robotics and Automation*, pages 995–1001, 2000.
- [26] Steven M La Valle. Motion planning. *IEEE Robotics & Automation Magazine*, 18(2):108–118, 2011.
- [27] Steven LaValle. Rapidly-exploring random trees: A new tool for path planning. *Research Report 9811*, 1998.
- [28] Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *arXiv preprint arXiv:1805.00909*, 2018.
- [29] Jiaoyang Li, Wheeler Ruml, and Sven Koenig. EECBS: A bounded-suboptimal search for multi-agent path finding. In *AAAI conference on artificial intelligence*, pages 12353–12362, 2021.
- [30] Jinhao Liang, Jacob K Christopher, Sven Koenig, and Ferdinando Fioretto. Simultaneous multi-robot motion planning with projected diffusion models. In *International Conference on Machine Learning*, pages 37162–37180, 2025.
- [31] Carlos E Luis, Marijan Vukosavljev, and Angela P Schoellig. Online trajectory generation with distributed model predictive control for multi-robot motion planning. *IEEE Robotics and Automation Letters*, 5(2):604–611, 2020.
- [32] Raghav Mishra and Ian R Manchester. EB-MBD: Emerging-barrier model-based diffusion for safe trajectory optimization in highly constrained environments. *arXiv preprint arXiv:2510.07700*, 2025.
- [33] Akmaral Moldagalieva, Joaquim Ortiz-Haro, Marc Toussaint, and Wolfgang Hönig. db-CBS: Discontinuity-bounded conflict-based search for multi-robot kinodynamic motion planning. In *International Conference on Robotics and Automation*, pages 14569–14575, 2024.
- [34] Keisuke Okumura. LaCAM: Search-based algorithm for quick multi-agent pathfinding. In *AAAI Conference on Artificial Intelligence*, pages 11655–11662, 2023.
- [35] Chaoyi Pan, Zeji Yi, Guanya Shi, and Guannan Qu. Model-based diffusion for trajectory optimization. In *Advances in Neural Information Processing Systems*, pages 57914–57943, 2024.
- [36] Jan Peters, Katharina Mulling, and Yasemin Altun. Relative entropy policy search. In *AAAI Conference on Artificial Intelligence*, pages 1607–1612, 2010.
- [37] Ralf Römer, Alexander von Rohr, and Angela P. Schoellig. Diffusion predictive control with constraints. In Necmiye Ozay, Laura Balzano, Dimitra Panagou, and Alessandro Abate, editors, *Learning for Dynamics & Control Conference*, pages 791–803, 2025.
- [38] Yorai Shaoul, Itamar Mishani, Shivam Vats, Jiaoyang Li, and Maxim Likhachev. Multi-robot motion planning with diffusion models. In *International Conference on Learning Representations*, 2025.
- [39] Guni Sharon, Roni Stern, Ariel Felner, and Nathan R Sturtevant. Conflict-based search for optimal multi-agent pathfinding. *Artificial intelligence*, 219:40–66, 2015.
- [40] Rahul Shome, Kiril Solovey, Andrew Dobson, Dan Halperin, and Kostas E Bekris. dRRT*: Scalable and informed asymptotically-optimal multi-robot motion planning. *Autonomous Robots*, 44(3):443–467, 2020.
- [41] Athanasios Sideris and James E Bobrow. An efficient sequential linear quadratic algorithm for solving nonlinear optimal control problems. In *American Control Conference*, pages 2275–2280, 2005.
- [42] Kiril Solovey, Oren Salzman, and Dan Halperin. Finding

a needle in an exponential haystack: Discrete RRT for exploration of implicit roadmaps in multi-robot motion planning. *International Journal of Robotics Research*, 35 (5):501–513, 2016.

- [43] Roni Stern, Nathan R. Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne T. Walker, Jiaoyang Li, Dor Atzmon, Liron Cohen, T. K. Satish Kumar, Roman Barták, and Eli Boyarski. Multi-agent pathfinding: Definitions, variants, and benchmarks. In *International Symposium on Combinatorial Search*, pages 151–158, 2019.
- [44] Marc Toussaint. Robot trajectory optimization using approximate inference. In *International Conference on Machine Learning*, pages 1049–1056, 2009.
- [45] Joe Watson and Jan Peters. Inferring smooth control: Monte carlo posterior policy iteration with gaussian processes. In *Conference on Robot Learning*, pages 67–79, 2023.
- [46] Joe Watson, Hany Abdulsamad, and Jan Peters. Stochastic optimal control as approximate input inference. In *Conference on Robot Learning*, pages 697–716, 2020.
- [47] Grady Williams, Paul Drews, Brian Goldfain, James M Rehg, and Evangelos A Theodorou. Information-theoretic model predictive control: Theory and applications to autonomous driving. *IEEE Transactions on Robotics*, 34(6):1603–1622, 2018.
- [48] Wei Xiao, Tsun-Hsuan Wang, Chuang Gan, Ramin Hasani, Mathias Lechner, and Daniela Rus. Safediffuser: Safe planning with diffusion probabilistic models. In *International Conference on Learning Representations*, 2023.
- [49] Haoru Xue, Chaoyi Pan, Zeji Yi, Guannan Qu, and Guanya Shi. Full-order sampling-based MPC for torque-level locomotion control via diffusion-style annealing. In *International Conference on Robotics and Automation*, pages 4974–4981, 2025.
- [50] Jichen Zhang, Liqun Zhao, Antonis Papachristodoulou, and Jack Umenberger. Constrained diffusers for safe planning and control. In *Advances in Neural Information Processing Systems*, pages 34965–34998, 2026.
- [51] Xiaohan Zhang, Xudong Mou, Rui Wang, Tianyu Wo, Ningbo Gu, Tiejun Wang, Cangbai Xu, and Xudong Liu. Robotdiffuse: Motion planning for redundant manipulator based on diffusion model. *arXiv preprint arXiv:2412.19500*, 2024.
- [52] Yuhao Zhang, Keisuke Okumura, Heedo Woo, Ajay Shankar, and Amanda Prorok. D4orm: Multi-robot trajectories with dynamics-aware diffusion denoised deformations. In *International Conference on Intelligent Robots and Systems*, pages 14118–14123, 2025.
- [53] Guangyao Zhou, Sivaramakrishnan Swaminathan, Rajkumar Vasudeva Raju, J. Swaroop Guntupalli, Wolfgang Lehrach, Joseph Ortiz, Antoine Dedieu, Miguel Lázaro-Gredilla, and Kevin Patrick Murphy. Diffusion model predictive control. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.