SPDD: GUIDED PROGRESSIVE DISTILLATION FOR FAST AND SAFE DIFFUSION-BASED PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models have shown strong performance in trajectory planning, but their iterative denoising process introduces considerable inference latency, which limits their deployment in real-time and resource-constrained settings. In this work, we propose safe progressive diffusion distillation (SPDD), a framework for accelerating diffusion-based planning while preserving teacher consistency and trajectory safety. SPDD introduces a quadratic program-based guidance mechanism that integrates prescribed-time control Lyapunov function (CLF) and control barrier function (CBF) into the denoising process. The CLF guides the student denoising process toward the teacher output, whereas the CBF is prioritized to enforce the desired safety constraints. We theoretically characterize the consistency and safety properties of the guided denoising process. To mitigate discretization errors caused by aggressive step reduction, SPDD progressively reduces the sampling horizon during distillation. Experiments on multiple trajectory planning tasks demonstrate that SPDD achieves competitive inference efficiency while maintaining low teacher–student discrepancy and improved safety compared with existing diffusion-based planning baselines.

1 INTRODUCTION

Diffusion models Ho et al. (2020) have achieved significant progress in robotics in recent years, particularly in trajectory planning and control tasks. Diffusion-based planners generate feasible trajectories by starting from random noise and iteratively performing reverse denoising steps Janner et al. (2022); Zhou et al. (2024). This iterative procedure requires multiple forward evaluations to produce a single trajectory, resulting in high inference latency. Simply reducing the number of denoising steps can accelerate sampling but often leads to degraded planning performance Kim et al. (2023).

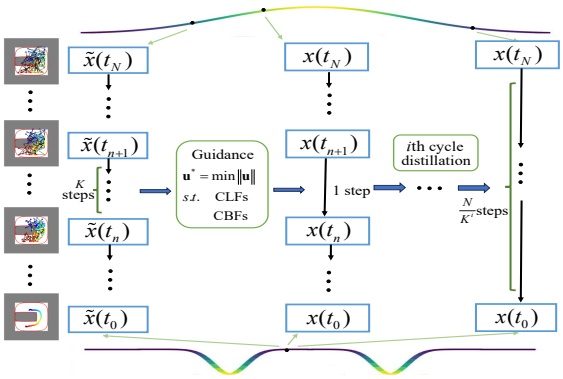


Figure 1: Overview of the proposed SPDD framework. In the teacher model, K sampling steps are required to transition from t_{n+1} to t_n, while the distilled student model performs this transition in a single step. Through i-th round of progressive distillation, the total number of sampling steps is reduced from NK to $\frac{N}{K^{i-1}}$.

Such slow inference suits the applicability of diffusion-based policies to tasks and scenarios that can tolerate long reaction times and high computational costs Peng et al. (2025); Yang et al. (2023). For example, relatively static manipulation tasks, such as pick-and-place, may allow slower inference. However, for high-frequency motion control and environments with dynamic obstacles or time-varying conditions, the inefficient sampling speed of diffusion models cannot support real-time replanning Chi et al. (2025); Chen et al. (2023). As a result, maintaining both responsiveness and safety during online robot operation becomes challenging Zhou et al. (2023). Given these challenges, our goal is to preserve the performance of diffusion-based policies while significantly reducing inference time. Specifically, we aim to develop an accelerated diffusion planning approach that

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

maintains the decision quality of the original diffusion policy and simultaneously provides safety guarantees for trajectory planning tasks.

Distillation techniques leverage a pretrained diffusion model to teach a new student model to take larger denoising steps, thereby reducing the number of sampling steps required during generation Song & Dhariwal (2023); Yin et al. (2024), and this approach has been widely studied in various generation domains, including image Liu et al. (2023b); Saharia et al. (2022), video Ho et al. (2022) and audio Liu et al. (2023a). In this work, we develop SPDD, which extends progressive diffusion distillation to safety-critical trajectory planning through control-theoretic guidance. Unlike existing acceleration methods that either reduce the sampling steps without retraining or distill the teacher output without explicitly enforcing safety, SPDD couples the denoising dynamics with a QP-based control Lyapunov function (CLF) Sontag (1983) and control barrier function (CBF) Ames et al. (2016) guidance mechanism. The CLF provides a principled way to regulate the deviation of the student output from the teacher reference, while the prioritized CBF preserves the safety requirement during denoising. We further analyze the resulting consistency and safety properties, including the consistency error introduced when the CLF constraint is relaxed. To make the guidance reliable under aggressive sampling-step reduction, we incorporate it into a progressive distillation procedure that gradually reduces the sampling horizon. This strategy ensures constraint satisfaction, and improves both the generation quality and safety of the distilled model.

2 Related work

Safe planning via generative model. In recent years, generative models, such as diffusion models and flow matching, have achieved widespread success not only in image generation but also in planning and control tasks. For trajectory planning problems, safety is a critical performance requirement. Several approaches have been proposed to generate trajectories that satisfy safety constraints. One line of work integrates safety considerations through constrained diffusion, where the denoising process is guided by cost functions or rejection sampling to ensure safety Carvalho et al. (2023); Mao et al. (2025); Kondo et al. (2024). However, these methods do not guarantee strict constraint satisfaction and may reduce sampling efficiency. One direction uses diffusion models directly as solvers for trajectory optimization to achieve safe planning based on the model Pan et al. (2024), where certain model is difficult to obtain. Another line of work Botteghi et al. (2023); Xiao et al. (2023) combines generative models with control-theoretic safety filters such as control barrier functions (CBFs), where generated trajectories are projected onto safe sets to provide formal guarantees. Römer et al. (2024) integrated a constrained optimal control layer into the denoising process to enforce state and action constraints. Although these strategies provide formal safety guarantees and preserves the flexibility of generative models, the additional optimization layer increases training complexity and introduces computational overhead, which can hinder real-time performance in high-dimensional robotic systems. Therefore, achieving strict safety guarantees while maintaining real-time efficiency and trajectory diversity remains an open challenge, motivating our research on safe planning via efficient generative models.

Efficient diffusion model. Although diffusion models have achieved remarkable results in generative tasks, their long Markovian sampling chains and high computational cost still limit practical deployment. Existing work to accelerate sampling can be grouped into three categories. 1) Algorithmic improvements to the sampling procedure reduce inference time without retraining the backbone model. Non-Markovian samplers such as DDIM Song et al. (2020a) allow training with many denoising steps while using fewer steps at inference to speed up sampling. However, empirical evidence shows that even with variable step schemes, directly reducing the number of denoising steps typically degrades sample quality Karras et al. (2022). 2) Structural changes to the diffusion model aim to improve sampling efficiency. Zhao et al. (2024) proposed Dynamic Diffusion Transformer to reduce inference time via adapt computation dynamically across time steps and spatial dimensions during generation. ParaDiGMS Shih et al. (2023) also seek to increase throughput via parallelization. Nevertheless, these approaches often demand significant computational resources, which limits the application in resource-constrained platforms. 3) Knowledge-distillation-based methods train a compact student model to approximate a teacher with far fewer sampling steps. Representative works include progressive distillation Salimans & Ho (2022), which iteratively refactors a DDIM model into student models with half the sampling steps. Additionally, consistency distillation Song et al. (2023); Prasad et al. (2024), which trains a student to reproduce the teacher's outputs at different

points along the same ODE trajectory. Despite these advances, how to formally guarantee output and trajectory consistency between student and teacher models, and ensuring safety during distillation and student inference in safety-critical tasks such as trajectory planning remain unresolved.

3 PRELIMINARIES AND PROBLEM SETTING

Diffusion Probability Models. Diffusion probabilistic models consist of two processes: a forward diffusion process and a reverse denoising process Song et al. (2020b); Ho et al. (2020). The forward diffusion process is defined over a continuous time horizon $t \in [0, T]$ by trajectories $\{x(t)\}_{t=0}^{T}$, which gradually perturbs data with Gaussian noise. Starting from data samples $x(0) \sim p_0$. The process evolves toward a tractable prior distribution, typically a Gaussian distribution p_T, such that $x(T) \sim p_T$. The reverse denoising process transforms samples drawn from the prior p_T back to the unknown data distribution p_0. For all diffusion processes, there exists a corresponding deterministic process whose trajectories share the same marginal probability densities $\{p_t(x)\}_{t=0}^{T}$. The generative objective is to learn a parameterized densities $\{p_t^{\theta}(x)\}_{t=0}^{T}$ that approximates the true denoising process. Then, the evolution of samples are formulated as a probability flow ordinary differential equation (PFODE) Song et al. (2020b).

$$\dot{x} = f(x, t) - \frac{1}{2} g(t)^2 \nabla_x \log p_t^{\theta}(x), \tag{1}$$

where $f(x, t)$ is a vector-valued function called the drift coefficient of $x(t)$, and $g(t)$ is a scalar function known as the diffusion coefficient of $x(t)$.

Diffusion models generate outputs through a sequential denoising process starting from an initial noisy state, and then numerically solves equation 1 by a solver-based method with a time schedule $\Gamma(N) = \{t_0 = 0, t_1, t_2, \cdots, t_N = T\}$ Zhang & Chen (2022); Lu et al. (2022). This iterative procedure requires multiple sampling steps, resulting in long inference times and significant computational overhead. To address this issue, distillation-based method is proposed.

Distillation-based Diffusion Sampling. Distillation techniques use a pre-trained teacher diffusion model to teach a new student model how to take larger denoising steps, reducing the total number of function evaluations required for generation. This basic framework behind distillation-based methods is illustrated in Fig. 1. Specifically, the teacher model and student model start from $\tilde{x}(t_{n+1})$ and $x(t_{n+1})$ $(0 \leq n \leq N-1)$, respectively, the sample process is written as:

$$\text{Teacher: } \tilde{x}(t_n) = \text{Solver}(\tilde{x}(t_{n+1}), K; \theta_t), \tag{2}$$

$$\text{Student: } x(t_n) = \text{Solver}(x(t_{n+1}), 1; \theta_s), \tag{3}$$

where K is the number of teacher sampling steps taken from t_{n+1} to t_n, θ_t and θ_s are the parameters of the teacher and student models, respectively. The student model θ_s is updated with the calculated loss function $\mathcal{L}(\theta_s) = d(x(t_n), \tilde{x}(t_n))$ using a distance metric $d(\cdot, \cdot)$ Salimans & Ho (2022); Song et al. (2023); Kim et al. (2023).

For the teacher model, the final output is required to satisfy safety constraints, e.g., $h(\tilde{x}(t_0)) \geq 0$, as well as other task-specific constraints such as those arising in trajectory planning. After training, the sampling process is performed with fixed parameters θ_t to provide reference signals for student model.

To ensure that the student model replicates the inference behavior of the multi-step teacher model, the student is trained such that its predicted output matches the teacher's result when starting from any timestep n. In particular, the student model is required to produce the same final sample, i.e., $x(t_0) = \tilde{x}(t_0)$, for inputs initialized at arbitrary intermediate timesteps.

Problem definition. Although distillation-based approaches significantly reduce the number of sampling steps, they introduce additional constraints.

1) **Consistency constraint**. The student model produces the same denoised output from any pair of points (x_i, t_i) and (x_j, t_j), $i, j \in [1, N-1]$ lying on the same denoising process with function $f_\theta(x_t, t) \mapsto x(t_0)$, and converges to fixed final sample of teacher model, i.e., $f_\theta(x_i, t_i) = f_\theta(x_j, t_j) = \tilde{x}(t_0)$.

2) **Safety constraint**. The student model is also required to ensure that its final output $x(t_0)$ satisfies the safety constraints $h(x(t_0)) \geq 0$, consistent with the teacher model.

Existing diffusion-based approaches typically incorporate these constraints into the training objective by augmenting the loss function. However, enforcing constraints in a soft manner through loss penalties does not provide guarantees that the generated samples will strictly satisfy the desired conditions. As a result, constraint violations may still occur at inference time.

Objective. The objective of this work is to develop a distillation-based diffusion framework that achieves fast sampling while explicitly guaranteeing that the final outputs of the student model satisfy the consistency constraint and safety constraint.

4 METHOD

In this section, we provide a guided distillation framework with formal guarantees for consistency and safety satisfaction. The section is organized as follows: (1) We formulate a control-theoretic guidance condition for constraints. (2) We detail how the proposed guidance condition is incorporated into the sampling process. (3) We provide theoretical analysis along with the complete algorithmic procedure.

4.1 CONTROL THEORY FOR GUIDANCE CONDITION

When conditional information is incorporated into the denoising process of a diffusion model, an additional guidance term can be introduced to steer the sampling dynamics. Under such guidance, the PFODE for student model can be modified by adding a control input $u(t)$ Dhariwal & Nichol (2021); Ajay et al. (2022); Römer et al. (2024), yielding

$$\dot{x} = f(x,t) - \frac{1}{2} g(t)^2 \nabla_x \log p_t^\theta(x) + u(t), \tag{4}$$

where $u(t)$ is the auxiliary guidance signal. It enables the diffusion trajectory to be actively steered toward satisfying desired constraints while preserving the underlying generative structure. In this subsection, we describe how the constraints are mapped into the guidance term.

While diffusion model denoising is originally defined in reverse time ($t_N \to t_0$), for the convenience of analysis, we reparameterize the denoising process in forward normalized time $t \in [0,1]$, where $t = 0$ is the initial time. This transformation does not affect the underlying dynamics.

From consistency constraint to Lyapunov stability. The consistency constraint requires the student model denoising trajectory converges to teacher model trajectory $\tilde{x}$ within a fixed timestep, while maintaining bounded deviation and avoiding divergence. This objective can be interpreted as stabilizing a dynamical system toward a desired reference trajectory. Such convergence properties can be established by constructing an appropriate Lyapunov function $V(x) = \|x - \tilde{x}(1)\|^2$, where $\tilde{x}(1)$ is a fixed terminal output of teacher model denoising process. To ensure convergence within a predefined time horizon, the evolution of the Lyapunov function is constrained to satisfy

$$\frac{\mathrm{d}V}{\mathrm{d}x}\left(f(x,t) - \frac{1}{2} g(t)^2 \nabla_x \log p_t^\theta(x) + u(t)\right) \leq -\alpha(t) V(x), \tag{5}$$

where $\alpha(t)$ is a piecewise time-varying function with predefined time $T_p \in (0,1)$ and a tunable parameter $c > 0$.

$$\alpha(t) = \begin{cases} \frac{c}{T_p - t}, & t \in [0, T_p) \\ c, & t \in [T_p, 1] \end{cases}$$

This condition enforces a time-varying decay rate on the Lyapunov function, ensuring that the student trajectory converges to the teacher trajectory as $t \to T_p^-$. When $t \geq T_p$, a standard exponential decay condition is imposed to maintain $V = 0$, such that the student denoising trajectory remains consistent with the teacher trajectory thereafter. (The proof is given in Appendix B.1).

From safety constraint to forward invariance. The safety constraint requires the final output satisfies $h(x) \geq 0$ even when the initial sample $h(x_N)$ may violate the constraint, i.e., $h(x_N) < 0$. This objective can be interpreted within the framework of prescribed-time forward invariance Abel et al. (2023). To achieve this property within a predefined time horizon, we employ a prescribed-time control barrier function (PT-CBF), which provides sufficient conditions for enforcing the sampling

trajectory satisfies $h(x) \geq 0$ within a predefined time. Given the safety constraint function $h(x)$, if the corresponding samples satisfy the following certificate

$$\frac{\mathrm{d}h}{\mathrm{d}x}\left(f(x,t) - \frac{1}{2}g(t)^2\nabla_x \log p_t^\theta(x) + u(t)\right) + \alpha(t)h(x) \geq 0, \tag{6}$$

the sampling will satisfy $h(x) \geq 0, \forall t \geq T_p$ (The proof is given in Appendix B.2).

4.2 Guidance design and theoretical guarantees

To simultaneously enforce the consistency and safety requirements, we compute the guidance term by formulating a quadratic program (QP) that incorporates both constraints. A key advantage of the QP formulation is that it unifies the consistency objective with the safety constraint within a single optimization framework Dawson et al. (2022); So & Fan (2023); Sun et al. (2021).

To guarantee feasibility of the optimization problem, the consistency condition equation 5 is treated as a soft constraint by introducing a relaxation variable, while the safety condition equation 6 is enforced as a hard constraint. By penalizing the relaxation term in the objective function, the QP balances consistency with safety while ensuring that the safety constraint is strictly satisfied. Specifically, the guidance is obtained by solving the following QP problem, we establish the following theoretical result to guarantee both consistency and safety constraints.[1]

$$\begin{aligned}
\mathbf{u}^* = &\arg\min_{\mathbf{u}=(u,\delta)} \|\mathbf{u}\|^2 + \frac{\rho}{2}\delta^2 \\
s.t. \quad &\frac{\mathrm{d}V}{\mathrm{d}x}\left(f(x,t) - \frac{1}{2}g(t)^2\nabla_x \log p_t^\theta(x) + u(t)\right) + \alpha(t)V(x) - \delta \leq 0 \\
&\frac{\mathrm{d}h}{\mathrm{d}x}\left(f(x,t) - \frac{1}{2}g(t)^2\nabla_x \log p_t^\theta(x) + u(t)\right) + \alpha(t)h(x) \geq 0
\end{aligned} \tag{7}$$

where δ is a relaxation parameter that ensures solvability of the QP, $\rho > 0$ is the penalty coefficient.

Theorem 1 (Consistency and safety under relaxed CLF-CBF guidance) *When the CLF-based convergence requirement toward $\tilde{x}(1)$ and the CBF-based safety requirement are jointly feasible, i.e., the relaxation satisfies $\delta(t) \equiv 0$ at each sampling time. In this case, the guidance input generated by the QP in equation 7 guarantees the student output satisfies both the consistency constraint and the safety constraint. When $\delta(t) \not\equiv 0$, the student output is guaranteed to achieve safety and slightly deviation from $\tilde{x}(1)$, and its residual error is bounded according to the accumulated relaxation term.*

4.3 Progressive distillation framework

At each timestep t_n, the QP equation 7 is solved to obtain the guidance. In principle, computing the guidance from the PFODE corresponds to a continuous-time optimization problem. However, in practice, the optimization is performed only at discrete sampling times, which effectively discretizes the continuous-time formulation. Specifically, over each sampling interval $[t_n, t_{n+1})$, the x_n and $u(t_n)$ are assumed to remain constant. This discretization introduces inter-sampling effects, meaning that constraint satisfaction is only explicitly enforced at the sampling instants Yang et al. (2020); Breeden et al. (2021); Niu et al. (2024). As the sampling

Algorithm 1: Progressive Distillation with QP-Guided Denoising

Input: Teacher model with parameters θ_t and steps N_t; distillation schedule $\{N_i\}_{i=1}^M$ with $N_M < \cdots < N_1 = N_t$; safety function $h(\cdot)$
Output: Sampling of student model

```
Initialize student model θ_s ← θ_t
for i = 1, 2, ..., M do
    Set sampling steps to N_i
    for each iteration do
        Perform the forward process
        for n = N_i, N_i − 1, ..., 1 do
            Formulate and solve the QP in equation 7 to obtain the guidance
            Update the denoising output via equation 4
        end for
    end for
    Update the teacher model: θ_t ← θ_s
end for
```

[1] A proof of the theorem is provided in Appendix B.3, and the closed-form solution is provided to improve efficiency of solving guidance.

interval increases, the deviation between updates may grow, which increases the likelihood of violating the consistency or safety conditions [2].

The inter-sampling effects becomes more pronounced in the distillation setting, where the student model aims to perform sampling with very few denoising steps, such as single-step or few-step generation. In this case, the time between successive QP updates becomes large, and the consistency and safety guarantees may no longer hold. To address this limitation, we adopt a progressive distillation strategy, which gradually reduces the number of sampling steps while preserving constraint satisfaction. We summarized the algorithm in Alg. 1.

5 EXPERIMENTS

We design experiments to answer the following questions: How does our distillation approach compare to the naive strategy of directly reducing sampling steps in terms of task performance and sample quality? Can safety guarantees be maintained throughout the distillation process? Finally, does the resulting student model reproduce the teacher's outputs while admitting formal theoretical guarantees of consistency and safety?

5.1 EXPERIMENT SETTING

We evaluate our method's advantages in safety and inference speed on four benchmark tasks: U-maze navigation, two locomotion tasks (Hopper and Walk2d) Fu et al. (2020), and a robotic manipulation task Janner et al. (2022). [3]

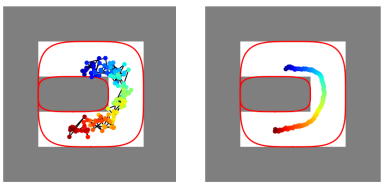
Metrics: We evaluate safety via the minimum value of safety constraint function when $t \geq T_p$, i.e., $\min h(x(t)), t \geq T_p$, a trajectory is considered safe when the value is positive. Inference efficiency is measured by the inference runtime during trajectory generation. To assess consistency between teacher and student outputs we use two measures: average trajectory error (ATE) and final trajectory error (FTE). Trajectory performance is measured by the D4RL normalized total reward and a stacked binary success reward.

Baselines: We compare our method with the following generative planning method: (1) **SafeDiffuser** Xiao et al. (2023), (2) **DDIM** Song et al. (2020a), (3) **Classic progressive distillation (PD) without guidance** Salimans & Ho (2022), (4) **Consistency distillation (CD)** Prasad et al. (2024).

5.2 PLANNING PERFORMANCE IN BENCHMARKS

Comparison with teacher model. We adopt SafeDiffuser with 1280 sampling steps as the teacher model. As shown in Table 1, directly reducing the number of sampling steps decreases inference time but leads to a substantial degradation in trajectory performance, as reflected by the reward. This degradation is expected, as fewer denoising steps result in a larger discretization gap between intermediate states, making it more difficult for the model to accurately approximate the reverse diffusion process.

Moreover, since SafeDiffuser only provides finite-time safety guarantees through step-wise CBF constraints, using too few sampling steps without properly adjusting the CBF parameters can lead to violations of safety constraints. This issue is particularly evident in the Umaze task when the sampling steps are reduced to 160, where unsafe trajectories may be generated. In contrast, our proposed distillation method effectively mitigates these issues, as shown in Fig. 2. Even with a significantly reduced number of sampling steps, the distilled model maintains trajectory performance comparable to that of the teacher model. This can be attributed to the progressive distillation strategy and



(a) Safediffuser (b) SPDD

Figure 2: Effect of the number of sampling steps on the safety and performance of trajectories.

[2]The detailed impact of the sampling interval on constraint satisfaction is analyzed in Appendix B.4

[3]The details of experiment are provided in Appendix C.1.

Table 1: Comparison of the teacher SafeDiffuser model with 1280 sampling steps, SafeDiffuser with directly reduced sampling steps, and SPDD across multiple tasks. Results are reported under different sampling steps; reward is presented as mean $\pm$ standard deviation, and inference time is reported in milliseconds.

Experiment	Metric	Teacher	SafeDiffuser			SPDD		
			640	320	160	640	320	160
Maze	Safety	0.12	0.07	0.05	-0.01	0.12	0.12	**0.13**
	Reward	$\mathbf{1.86 \pm 0.05}$	1.16 ± 0.01	1.02 ± 0.10	0.71 ± 0.01	1.30 ± 0.02	1.21 ± 0.03	1.07 ± 0.03
	Inf. time	121.64	96.23	85.42	69.28	89.92	70.64	**61.93**
Hopper	Safety	0.02	0.00	0.00	0.00	0.00	0.00	0.00
	Reward	$\mathbf{1.02 \pm 0.02}$	0.47 ± 0.09	0.16 ± 0.08	0.11 ± 0.02	0.49 ± 0.03	0.45 ± 0.00	0.39 ± 0.02
	Inf. time	96.84	72.75	49.02	22.71	34.27	25.41	**19.83**
Walk2d	Safety	0.03	0.00	0.00	0.00	0.00	0.00	0.00
	Reward	$\mathbf{0.65 \pm 0.04}$	0.34 ± 0.12	0.22 ± 0.05	0.17 ± 0.01	0.44 ± 0.02	0.36 ± 0.09	0.35 ± 0.01
	Inf. time	121.04	92.04	79.12	42.54	42.99	35.10	**30.08**
Manipulation	Safety	0.02	0.00	0.00	-0.00	0.00	0.00	0.00
	Reward	$\mathbf{0.82 \pm 0.05}$	0.63 ± 0.06	0.58 ± 0.10	0.35 ± 0.08	0.77 ± 0.02	0.65 ± 0.02	0.60 ± 0.07
	Inf. time	98.43	81.32	65.24	50.79	45.33	39.28	**29.20**

prescribed-time safety, where the optimization-based guidance enforces consistency with the teacher while integrating prescribed-time CLF-CBF constraints. As a result, the model is able to preserve both the fidelity of the generated trajectories and adherence to safety constraints.

Overall, these results answer our first two research questions: the proposed method achieves substantial acceleration without sacrificing task performance, and the distilled model continues to satisfy safety requirements under reduced sampling steps, demonstrating both efficiency and reliability in safety-critical planning scenarios.

Comparison with other efficient diffusion model. In Table 2, although DDIM and PD can achieve slightly shorter inference time in some settings, they do not explicitly enforce consistency or safety during the reduced-step sampling process. CD improves the consistency of one-step or few-step generation through direct distillation, but it does not incorporate explicit safety constraints. In contrast, SPDD combines progressive distillation with CLF-CBF-guided denoising, achieving a favorable trade-off between inference efficiency, teacher consistency, and safety. The safety values of SPDD remain nonnegative across the evaluated tasks and sampling budgets, whereas the other diffusion-based baselines may exhibit negative safety values or larger consistency errors. These results demonstrate that SPDD can accelerate diffusion-based trajectory generation while preserving both the quality and safety of the generated trajectories.

Table 2: Performance comparison of ATE, FTE, inference time and safety function value between consistency distillation, classic progressive distillation without guidance (PD) and SPDD across datasets. For DDIM, we only report the error between its final output and the output of DDPM as the FTE, while excluding ATE from the comparison.

Experiment	Metric	DDIM			CD			PD			SPPD		
		320	160	80	320	160	80	320	160	80	320	160	80
Maze	ATE	—	—	—	1.43	1.48	1.57	1.51	1.59	1.67	**1.36**	1.58	1.73
	FTE	0.22	0.24	0.24	0.20	0.21	0.25	0.25	0.34	0.42	0.08	**0.07**	**0.07**
	Inf. time	76.53	58.92	50.26	71.43	60.94	50.35	68.23	58.38	**55.21**	70.64	61.93	49.26
	safety	0.05	0.08	0.05	0.06	0.03	0.04	0.05	0.03	0	0.12	**0.13**	**0.13**
Hopper	ATE	—	—	—	0.15	0.23	0.25	0.23	0.35	0.48	**0.11**	0.18	0.21
	FTE	0.08	0.07	0.08	0.06	0.08	0.06	0.09	0.12	0.15	**0.05**	0.06	**0.05**
	Inf. time	32.56	21.74	18.21	24.23	18.72	15.25	21.46	16.15	**13.10**	25.41	19.83	16.52
	safety	0.00	-0.00	-0.00	-0.00	-0.00	0.00	-0.00	-0.00	-0.01	0.00	0.00	0.00
Walk2d	ATE	—	—	—	0.28	0.29	0.35	0.34	0.42	0.48	**0.18**	0.21	0.25
	FTE	0.09	0.08	0.08	0.08	0.09	0.08	0.09	0.12	0.13	0.07	0.07	**0.06**
	Inf. time	42.43	38.95	30.94	36.90	30.97	23.92	34.17	28.05	**20.89**	35.10	30.08	23.21
	safety	0.00	-0.01	-0.01	-0.01	-0.00	-0.00	-0.00	-0.00	-0.01	0.00	0.00	0.00
Manipulation	ATE	—	—	—	**0.31**	0.35	0.38	0.38	0.42	0.51	**0.25**	0.29	0.31
	FTE	0.23	0.25	0.29	0.18	0.20	0.21	0.18	0.27	0.30	**0.08**	**0.08**	0.09
	Inf. time	40.33	31.25	26.10	38.25	27.92	19.97	35.02	27.61	**14.32**	39.28	29.20	15.93
	safety	-0.01	-0.00	-0.00	-0.01	-0.00	-0.01	-0.00	0.00	-0.01	0.00	0.00	0.00

The impact of prescribed-time In Fig. 3, we analyze the impact of key parameters on the Hopper benchmark task. First, regarding the prescribed-time parameter T_p, the student model is guided to match the teacher trajectories within a normalized time horizon $[0, 1]$. As shown in Fig. 3, a shorter prescribed time generally leads to a smaller average trajectory error, since stronger guidance enforces closer alignment with the teacher. However, this comes with a trade-off: a shorter time horizon requires larger guidance magnitudes, which imposes stronger constraints on the generation process and limits exploration, resulting in lower rewards.

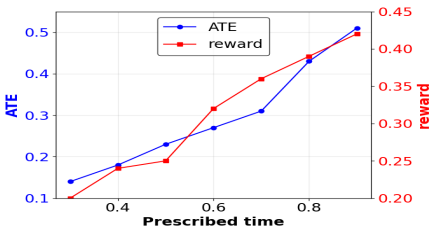


Figure 3: Effect of T_p on the trajectory performance.

The necessary of progressive distillation. In addition, we investigate the role of progressive distillation via Walker2d task. The motivation is to address the discretization error arising from incorporating continuous-time safety constraints into the discrete denoising process of diffusion models. In our formulation, the guidance term is derived from a continuous-time optimization problem under CLF-CBF constraints, but is implemented at each discrete denoising step via a QP. When the denoising step size becomes too large, i.e., overly aggressive step reduction, this discrete approximation becomes inaccurate, and the resulting QP may fail to faithfully capture the underlying continuous-time constraints. To validate this, we fix the final number of sampling steps to 160 and vary per-stage step-reduction factor $K = 2, 4, 8$. The results show that using more gradual step reduction (smaller K) leads to better consistency and safety, demonstrating the necessity of progressive distillation in maintaining performance under aggressive sampling reduction.

Table 3: Effect of the number of progressive distillation stages K on performance.

Metric	Safety	ATE	FTE↑	Reward↓
$K = 2$	0.00	0.21	0.07	0.36 ± 0.09
$K = 4$	0.00	0.23	0.07	0.32 ± 0.03
$K = 8$	0.00	0.31	0.09	0.28 ± 0.02

Ablation result. We compare direct distillation (DD), PD, PD with CLF-only guidance (PD-CLF-only), or with CBF-only guidance (PD-CBF-only), and the proposed SPDD in Table 4. In addition to ATE, FTE, and inference time, we report stricter safety statistics, including the safety rate and the minimum safety function value over multiple runs under the 160-step setting. These results provide a more comprehensive evaluation of safety-critical performance and further demonstrate the effectiveness of SPDD. In particular, they show that neither progressive distillation nor guidance alone is sufficient; rather, the combination of progressive distillation and CLF-CBF guidance yields the best trade-off among safety, teacher consistency, and inference efficiency.

In addition, the qualitative influence of the relaxation factor is illustrated through cases 1–3 of Umaze task in Fig. 4. To further quantify the behavior of the proposed method, we record the relaxation factor $\delta(t)$ during the denoising process and report its maximum value $\max \delta$, average value $\bar{\delta}$, activation frequency $\Pr(\delta > 0)$, and FTE under the 160-step setting. As shown in Table 5, the maximum and average values of δ remain small across all tasks. Moreover, $\Pr(\delta > 0)$ is low, indicating that the CLF and CBF constraints are compatible for most of the denoising process. Therefore, the relaxation factor is activated only when necessary to preserve the feasibility of the safety-constrained QP, rather than being continuously used to weaken the consistency requirement. In addition, the low FTE provides empirical evidence that the accumulated error introduced by the relaxed CLF remains small, and that the student output stays close to the corresponding teacher output in practice. In particular, although the relaxed CLF does not generally provide exact prescribed-time convergence, the observed relaxation statistics and FTE demonstrate that the resulting practical consistency error is negligible for the evaluated tasks.

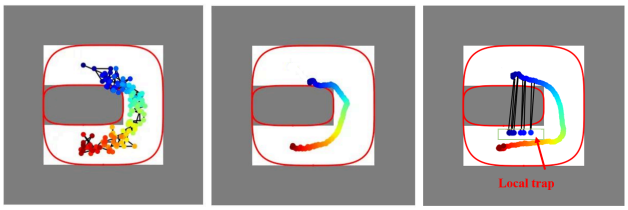
Table 5: Statistics of the CLF relaxation factor and final trajectory error for SPDD. Values of $\delta(t)$ smaller than 10^{-4} are treated as numerical zeros.

Experiment	$\max \delta$	$\bar{\delta}$	$\Pr(\delta > 0)$	FTE
Maze	0.002	0.002	0.6%	0.07
Hopper	0.001	0.000	0.6%	0.06
Walk2d	0.000	0.000	0	0.07
Manipulation	0.001	0.001	1.2%	0.08

Table 4: Comparison of DD, PD, PD with CLF-only or CBF-only guidance, and SPDD.

Experiment	Metric	DD	PD	PD-CLF-only	PD-CBF-only	SPDD
Maze	ATE	1.57 ± 0.04	1.56 ± 0.07	$\mathbf{1.53 \pm 0.02}$	1.58 ± 0.03	1.55 ± 0.03
	FTE	0.28 ± 0.04	0.34 ± 0.03	0.08 ± 0.00	0.33 ± 0.02	$\mathbf{0.07 \pm 0.01}$
	Inf. time	59.79 ± 3.49	$\mathbf{52.76 \pm 4.05}$	59.78 ± 3.46	59.84 ± 3.25	61.03 ± 0.91
	min. $h(x)$	-0.01	-0.02	-0.02	$\mathbf{0.13}$	0.11
	safety rate	96%	90%	90%	$\mathbf{100\%}$	$\mathbf{100\%}$
Hopper	ATE	0.23 ± 0.04	0.34 ± 0.03	0.21 ± 0.01	0.35 ± 0.02	$\mathbf{0.17 \pm 0.02}$
	FTE	0.09 ± 0.03	0.12 ± 0.01	0.06 ± 0.01	0.15 ± 0.02	$\mathbf{0.06 \pm 0.00}$
	Inf. time	18.45 ± 4.76	16.36 ± 1.96	19.03 ± 2.96	$\mathbf{18.28 \pm 3.23}$	18.52 ± 1.24
	min. $h(x)$	-0.01	-0.02	-0.03	$\mathbf{0.00}$	$\mathbf{0.00}$
	safety rate	98%	92%	91%	$\mathbf{100\%}$	$\mathbf{100\%}$
Walk2d	ATE	0.31 ± 0.02	0.43 ± 0.04	0.23 ± 0.02	0.40 ± 0.03	$\mathbf{0.21 \pm 0.01}$
	FTE	0.09 ± 0.00	0.12 ± 0.01	0.07 ± 0.00	0.12 ± 0.02	$\mathbf{0.07 \pm 0.01}$
	Inf. time	29.92 ± 1.91	$\mathbf{24.43 \pm 4.85}$	29.83 ± 3.87	31.05 ± 2.21	30.21 ± 1.05
	min. $h(x)$	-0.01	-0.03	-0.03	$\mathbf{0.01}$	0.00
	safety rate	96%	93%	92%	$\mathbf{100\%}$	$\mathbf{100\%}$
Manipulation	ATE	0.32 ± 0.02	0.38 ± 0.04	0.30 ± 0.02	0.43 ± 0.03	$\mathbf{0.28 \pm 0.03}$
	FTE	0.12 ± 0.03	0.28 ± 0.02	0.09 ± 0.00	0.24 ± 0.03	$\mathbf{0.07 \pm 0.01}$
	Inf. time	28.91 ± 1.43	$\mathbf{24.02 \pm 1.04}$	28.24 ± 2.93	26.74 ± 2.57	26.98 ± 3.44
	min. $h(x)$	-0.01	-0.03	-0.01	$\mathbf{0.00}$	$\mathbf{0.00}$
	safety rate	95%	92%	92%	$\mathbf{100\%}$	$\mathbf{100\%}$

Case 1: Hard-constrained CLF and CBF in QP. When solving QP, we do not introduce relaxation parameter δ. As a result, the QP may become infeasible in certain cases, yielding no valid guidance term. In such case, the distilled student model, which relies on fewer sampling steps, may fail to generate feasible trajectories.



(a) Case 1 (b) Case 2 (c) Case 3

Figure 4: Different QP formulations for guidance in the Umaze task.

Case 2: Soft-constrained CBF in QP. When solving QP, introducing δ in the CBF constraints relaxes the safety conditions, allowing constraint violations to improve feasibility. As a result, the guidance tends to prioritize consistency with the teacher model, driving the student to closely follow the teacher's trajectory. However, although the teacher model is trained with safety constraints, small deviations introduced during distillation can accumulate and lead to locally unsafe trajectories.

Case 3: No safety constraint in teacher model. When the teacher model does not incorporate safety constraints (e.g., a standard DDPM), QP-based guidance is used during distillation to enforce safety. However, under strong safety constraints, the intermediate samples can be overly restricted by the QP guidance and are unable to move toward high-likelihood regions of the diffusion model. This case may cause the trajectory to get trapped in local regions, as stated in Xiao et al. (2023), leading to local trapping issues in trajectory.

6 Conclusion

In this paper, we proposed a guidance-based distillation framework for diffusion-based trajectory planning that enables fast inference while preserving safety. Our method leverages a safety-constrained teacher diffusion model and introduces a QP-based guidance derived from prescribed-time CLF-CBF conditions, providing theoretical guarantees on both consistency with the teacher and safety during generation, without requiring additional loss terms. To address the challenges posed by large denoising steps, we further develop a progressive distillation strategy that gradually reduces the sampling horizon, mitigating discretization errors and improving stability. Experimental results demonstrate that the proposed approach significantly accelerates sampling while maintaining trajectory quality and safety, highlighting its effectiveness for real-time, safety-critical applications.

REFERENCES

Imoleayo Abel, Drew Steeves, Miroslav Krstić, and Mrdjan Janković. Prescribed-time safety design for strict-feedback nonlinear systems. *IEEE Transactions on Automatic Control*, 69(3):1464–1479, 2023.

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022.

Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8): 3861–3876, 2016.

Nicolò Botteghi, Federico Califano, Mannes Poel, and Christoph Brune. Trajectory generation, control, and safety with denoising diffusion probabilistic models. *arXiv preprint arXiv:2306.15512*, 2023.

Joseph Breeden, Kunal Garg, and Dimitra Panagou. Control barrier functions in sampled-data systems. *IEEE Control Systems Letters*, 6:367–372, 2021.

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 *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1916–1923. IEEE, 2023.

Shang-Fu Chen, Hsiang-Chun Wang, Ming-Hao Hsu, Chun-Mao Lai, and Shao-Hua Sun. Diffusion model-augmented behavioral cloning. *arXiv preprint arXiv:2302.13335*, 2023.

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. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Charles Dawson, Zengyi Qin, Sicun Gao, and Chuchu Fan. Safe nonlinear control using robust neural lyapunov-barrier functions. In *Conference on Robot Learning*, pp. 1724–1735. PMLR, 2022.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Caelan Reed Garrett, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Pddlstream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning. In *Proceedings of the international conference on automated planning and scheduling*, volume 30, pp. 440–448, 2020.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in neural information processing systems*, 35:8633–8646, 2022.

Michael Janner, Yilun Du, Joshua Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *International Conference on Machine Learning*, pp. 9902–9915. PMLR, 2022.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Dongjun Kim, Chieh-Hsin Lai, Wei-Hsiang Liao, Naoki Murata, Yuhta Takida, Toshimitsu Uesaka, Yutong He, Yuki Mitsufuji, and Stefano Ermon. Consistency trajectory models: Learning probability flow ode trajectory of diffusion. *arXiv preprint arXiv:2310.02279*, 2023.

Kota Kondo, Andrea Tagliabue, Xiaoyi Cai, Claudius Tewari, Olivia Garcia, Marcos Espitia-Alvarez, and Jonathan P How. Cgd: Constraint-guided diffusion policies for uav trajectory planning. *arXiv preprint arXiv:2405.01758*, 2024.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. *arXiv preprint arXiv:2301.12503*, 2023a.

Xingchao Liu, Xiwen Zhang, Jianzhu Ma, Jian Peng, et al. Instaflow: One step is enough for high-quality diffusion-based text-to-image generation. In *International Conference on Learning Representations*, 2023b.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022.

Wule Mao, Zhouheng Li, Yunhao Luo, Yilun Du, and Lei Xie. Rapid and safe trajectory planning over diverse scenes through diffusion composition. *arXiv preprint arXiv:2507.04384*, 2025.

Luyao Niu, Bhaskar Ramasubramanian, Andrew Clark, and Radha Poovendran. Sampling and quantization-aware control barrier functions for safety-critical control of cyber-physical systems. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pp. 1637–1644. IEEE, 2024.

Chaoyi Pan, Zeji Yi, Guanya Shi, and Guannan Qu. Model-based diffusion for trajectory optimization. *Advances in Neural Information Processing Systems*, 37:57914–57943, 2024.

Mingxing Peng, Kehua Chen, Xusen Guo, Qiming Zhang, Hui Zhong, Meixin Zhu, and Hai Yang. Diffusion models for intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 26:21526–21543, 2025.

Aaditya Prasad, Kevin Lin, Jimmy Wu, Linqi Zhou, and Jeannette Bohg. Consistency policy: Accelerated visuomotor policies via consistency distillation. *arXiv preprint arXiv:2405.07503*, 2024.

Ralf Römer, Alexander von Rohr, and Angela P Schoellig. Diffusion predictive control with constraints. *arXiv preprint arXiv:2412.09342*, 2024.

Chitwan Saharia, Jonathan Ho, William Chan, Tim Salimans, David J Fleet, and Mohammad Norouzi. Image super-resolution via iterative refinement. *IEEE transactions on pattern analysis and machine intelligence*, 45(4):4713–4726, 2022.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. *arXiv preprint arXiv:2202.00512*, 2022.

Andy Shih, Suneel Belkhale, Stefano Ermon, Dorsa Sadigh, and Nima Anari. Parallel sampling of diffusion models. *Advances in Neural Information Processing Systems*, 36:4263–4276, 2023.

Oswin So and Chuchu Fan. Solving stabilize-avoid optimal control via epigraph form and deep reinforcement learning. *arXiv preprint arXiv:2305.14154*, 2023.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020a.

Yang Song and Prafulla Dhariwal. Improved techniques for training consistency models. *arXiv preprint arXiv:2310.14189*, 2023.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020b.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. 2023.

Eduardo D Sontag. A lyapunov-like characterization of asymptotic controllability. *SIAM journal on control and optimization*, 21(3):462–471, 1983.

Dawei Sun, Susmit Jha, and Chuchu Fan. Learning certified control using contraction metric. In *conference on Robot Learning*, pp. 1519–1539. PMLR, 2021.

Wei Xiao, Tsun-Hsuan Wang, Chuang Gan, Ramin Hasani, Mathias Lechner, and Daniela Rus. Safediffuser: Safe planning with diffusion probabilistic models. In *The thirteenth international conference on learning representations*, 2023.

Guang Yang, Calin Belta, and Roberto Tron. Continuous-time signal temporal logic planning with control barrier functions. In *2020 American Control Conference (ACC)*, pp. 4612–4618. IEEE, 2020.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM computing surveys*, 56(4):1–39, 2023.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6613–6623, 2024.

Qinsheng Zhang and Yongxin Chen. Fast sampling of diffusion models with exponential integrator. *arXiv preprint arXiv:2204.13902*, 2022.

Wangbo Zhao, Yizeng Han, Jiasheng Tang, Kai Wang, Yibing Song, Gao Huang, Fan Wang, and Yang You. Dynamic diffusion transformer. *arXiv preprint arXiv:2410.03456*, 2024.

Guangyao Zhou, Sivaramakrishnan Swaminathan, Rajkumar Vasudeva Raju, J Swaroop Guntupalli, Wolfgang Lehrach, Joseph Ortiz, Antoine Dedieu, Miguel Lázaro-Gredilla, and Kevin Murphy. Diffusion model predictive control. *arXiv preprint arXiv:2410.05364*, 2024.

Siyuan Zhou, Yilun Du, Shun Zhang, Mengdi Xu, Yikang Shen, Wei Xiao, Dit-Yan Yeung, and Chuang Gan. Adaptive online replanning with diffusion models. *Advances in Neural Information Processing Systems*, 36:44000–44016, 2023.

AI USE STATEMENT

In this work, we used generative AI tools to improve the language, grammar, and overall readability of the manuscript. We did not use generative AI tools to generate research results, design experiments, develop the methodology, analyze data, or make scientific claims. All content generated or revised with AI assistance was reviewed and verified by the authors.

A PRELIMINARIES

Given a differentiable dynamical system $\dot{x} = f(x) + g(x)u$ with a target goal x^*, a safe constraint defined by a continuous differentiable function $h(x) \geq 0$ and corresponding safe set $C = \{h(x) \geq 0\}$, the following definitions are introduced.

Definition 1 (Stability) *The system is asymptotically stable if the state x approaches x^* within a small tolerance ϵ, i.e.,*

$$\lim_{t\to\infty} \|x - x^*\| \leq \epsilon.$$

Definition 2 (Forward invariance) *If for every $x(0) \in C$, the state $x(t)$ always stays in the set C for $t \geq 0$, the set C is forward invariant. Then the safety of system is guaranteed and the set C is called safe set.*

Definition 3 (Control Lyapunov function) *A continuously differentiable function $V(x)$ is an exponentially stabilizing control Lyapunov function if there exist positive constants $c_1, c_2, c_3 > 0$ such that for all $x = (x_1, x_2)$, the following inequalities hold,*

$$c_1\|x_1\|^2 \leq V(x) \leq c_2\|x_2\|^2$$
$$\frac{dV}{dx}(f(x) + g(x)u) + c_3V(x) \leq 0.$$

Lemma 1 *If $V(x)$ is a control Lyapunov function, then the control input $u(t) \in \{u : \frac{dV}{dx}(f(x) + g(x)u) + c_3V(x) \leq 0\}$ renders the system asymptotically stable.*

Definition 4 (Control barrier function) *Given a set C, $h(x)$ is a control barrier function (CBF) for system if there exists a class $\mathcal{K}$ function $\alpha(\cdot)$ such that*

$$\frac{dh}{dx}(f(x) + g(x)u) + \alpha(h(x)) \geq 0, \forall x \in C.$$

Lemma 2 *If $h(x)$ is a control barrier function, then Lipschitz continuous control input $u(t) \in \{u : \frac{dh}{dx}(f(x) + g(x)u) + \alpha(h(x)) \geq 0\}$ renders the set C forward invariant.*

For the QP equation 7, some assumptions are required. First, the Lyapunov function V is differentiable and $\frac{\mathrm{d}V}{\mathrm{d}x} \neq 0$. Second, the differentiable function $h(x)$ satisfies the relative degree one condition, i.e., $\frac{\mathrm{d}h}{\mathrm{d}x} \neq 0$. According to the definition of Lyapunov function, it is clear that the function satisfies the assumption. For the safety function $h(x)$, it is defined as signed distance function, and the details about the definition of safety is introduced in Appendix C later. In general, signed distance functions are not globally smooth. Nevertheless, under mild conditions, such functions are differentiable almost everywhere for geometric sets with smooth boundaries. For non-smooth boundaries, this issue can be effectively addressed by decomposing the safety region and supplementing additional safety constraints. Therefore, the differentiability assumption is only adopted to facilitate theoretical derivation and enhance mathematical rigor, without bringing essential restrictions to practical applications. In addition, this work mainly focuses on the scenario with relative degree one. For safety constraints with higher relative degrees, high-order control barrier functions can be directly utilized to ensure system safety.

B Theoretical Guarantees

B.1 The proof for the prescribed-time stability

Solving the differential inequality equation 5 gives

$$V(t) \leq V(t_0)e^{-\int_{t_0}^{t} \alpha(s)ds},$$

since $\alpha(t) = \frac{c}{T_p - t}$ for $t \in [t_0, T_p)$, we have $\int_{t_0}^{t} \alpha(s)ds = c(\ln \frac{T_p}{T_p - t} - \ln \frac{T_p}{T_p - t_0})$. Therefore, the following inequality holds

$$V \leq V(t_0)\big(\frac{T_p - t}{T_p - t_0}\big)^c$$

which gives

$$\lim_{t \to T_p^-} V(t) = 0.$$

From the stability condition derived in equation 5, we derive that the Lyapunov function asymptotically converges to zero when $t \to T_p^-$. When $t \geq T_p$, equation 5 is a classic CLF condition. Then, according to comparison principle and $V \geq 0$, we have

$$V(t) \leq V(T_p)e^{-c(t - T_p)} = 0$$

Therefore, the prescribed-time stability property is fully proven.

B.2 The proof for the prescribed-time safety

When the initial safety constraint is unsatisfied, i.e., $h(t_0) < 0$, solving the differential inequality equation 6 gives

$$h(t) \geq e^{-\int_{t_0}^{T_p} \alpha(s)ds} h(t_0)$$

where $\int_{t_0}^{T_p} \alpha(s)ds = c(\ln \frac{T_p}{T_p - t} - \ln \frac{T_p}{T_p - t_0})$. Since we have

$$\lim_{t \to T_p^-} e^{-\int_{t_0}^{T_p} \alpha(s)ds} = \lim_{t \to T_p^-} (\frac{T_p - t}{T_p - t_0})^c = 0$$

then

$$\lim_{t \to T_p^-} h(t) = 0$$

This implies that $h(x)$ will converge to the safe set $C = \{h(x) \geq 0\}$ as $t \to T_p$. When $t \geq T_p$, the safety condition equation 6 coincides with the standard CBF condition. According to Lemma 2 in Appendix A, the safety constraint $h(x) \geq 0$ holds for all $t \geq T_p$. Therefore, the prescribed-time safety property is fully proven.

B.3 The proof for Theorem 1

In QP equation 7, the CLF constraint in equation 7 satisfies

$$\dot{V}(x,t) + \alpha(t)V(x,t) \leq \delta(t), \qquad \delta(t) \geq 0.$$

The CBF constraint is imposed as a hard constraint and satisfies

$$\dot{h}(x,t) + \alpha(t)h(x,t) \geq 0.$$

First, consider the case in which the CLF-based convergence requirement and the CBF-based safety requirement are jointly feasible. In this case, the zero-relaxation solution is feasible at every sampling time, namely, $\delta(t) \equiv 0$. Consequently, the CLF condition reduces to

$$\dot{V}(x,t) + \alpha(t)V(x,t) \leq 0.$$

According to Appendix B.1, the student output satisfies the consistency constraint within the prescribed time T_p. Moreover, because the CBF constraint is enforced as a hard constraint, the prescribed-time forward-invariance condition holds. Under the prescribed-time CBF condition and Appendix B.2, the resulting student trajectory satisfies

$$h(x(t)) \geq 0, \qquad t \geq T_p.$$

Thus, when $\delta(t) \equiv 0$, both consistency and safety are guaranteed.

Next, consider the case in which the CLF and CBF requirements are incompatible at some sampling times. To preserve QP feasibility, the CLF constraint is relaxed, leading to

$$\dot{V}(x,t) + \alpha(t)V(x,t) \leq \delta(t), \qquad \delta(t) \not\equiv 0.$$

integrating from t_0 to t yields

$$V(x,t) \leq e^{-\int_{t_0}^{t} \alpha(\tau)\, d\tau} V(x,t_0) + \int_{t_0}^{t} e^{-\int_{s}^{t} \alpha(\tau)\, d\tau} \delta(s)\, ds.$$

The first term represents the nominal CLF-induced convergence, while the second term represents the accumulated consistency error caused by the relaxation. Therefore, the student output remains close to the teacher output when the relaxation magnitude and its accumulated effect are small, but exact consistency cannot generally be guaranteed when $\delta(t) \not\equiv 0$.

Finally, since the CBF constraint remains hard in both cases, the safety condition continues to be enforced provided that the QP is feasible and the assumptions required by the prescribed-time CBF condition hold. Therefore, the QP-guided student output satisfies safety in both cases, while the consistency guarantee is exact when $\delta(t) \equiv 0$ and becomes a bounded practical consistency guarantee when $\delta(t) \not\equiv 0$.

Moreover, QP has a closed-form expression for $\boldsymbol{u}^*$. Define

$$y_1(x) = \left[\frac{\mathrm{d}V}{\mathrm{d}x}, -1\right]^T, \quad p_1(x) = -\frac{\mathrm{d}V}{\mathrm{d}x}\left(f(x,t) - \frac{1}{2}g(t)^2 \nabla_x \log p_t^\theta(x)\right) - \alpha(t)V(x),$$

$$y_2(x) = \left[-\frac{\mathrm{d}h}{\mathrm{d}x}, 0\right]^T, \; p_2(x) = \frac{\mathrm{d}h}{\mathrm{d}x}\left(f(x,t) - \frac{1}{2}g(t)^2 \nabla_x \log p_t^\theta(x)\right) + \alpha(t)h(x),$$

Let $G(x) = [G_{ij}(x)] = [y_i(x)^T y_j(x)], i, j = 1, 2$ be the Gram matrix. Due to the linear independence of $\{y_1(x), y_2(x)\}$, $G(x)$ is positive definite. According to KKT condition, the solution to QP equation 7 is

$$\boldsymbol{u}^*(x) = \lambda_1(x)y_1(x) + \lambda_2(x)y_2(x),$$

where $\lambda = [\lambda_1(x), \lambda_2(x)]^T$ is the unique solution to

$$\begin{aligned} G(x)\lambda(x) &\leq p(x), \\ \lambda(x) &\leq 0, \\ [G(x)\lambda(x)]_i < p_i(x) &\rightarrow \lambda_i(x) = 0, \end{aligned}$$

where $[\cdot]_i$ is the i-th row of the quantity in brackets, and the inequalities hold componentwise. Define a Lipschitz continuous function

$$\omega(r) = \begin{cases} 0, & \text{if } r > 0 \\ r, & \text{if } r \leq 0 \end{cases}$$

Then, λ_1 and λ_2 can be experssed as

If $G_{21}(x)\omega(p_2(x)) - G_{22}(x)p_1(x) < 0$:

$$\begin{bmatrix} \lambda_1(x) \\ \lambda_2(x) \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{\omega(p_2(x))}{G_{22}} \end{bmatrix}$$

else if $G_{12}(x)\omega(p_1(x)) - G_{11}(x)p_2(x) < 0$:

$$\begin{bmatrix} \lambda_1(x) \\ \lambda_2(x) \end{bmatrix} = \begin{bmatrix} \frac{\omega(p_1(x))}{G_{11}} \\ 0 \end{bmatrix}$$

otherwise:

$$\begin{bmatrix} \lambda_1(x) \\ \lambda_2(x) \end{bmatrix} = \begin{bmatrix} \frac{\omega(G_{22}p_1(x) - G_{21}(x)p_2(x))}{G_{11}(x)G_{22}(x) - G_{12}(x)G_{21}(x)} \\ \frac{\omega(G_{11}p_2(x) - G_{12}(x)p_1(x))}{G_{11}(x)G_{22}(x) - G_{12}(x)G_{21}(x)} \end{bmatrix}$$

Since $G(x)$ is positive definite, $G_{11}(x)G_{22}(x) - G_{12}(x)G_{21}(x) > 0$. Because the standard properties for the composition and product of locally Lipschitz continuous functions, each of the expressions for λ_1 and λ_2 are locally Lipschitz continuous. completing the proof.

B.4 The impact of step length for QP

The denoising process described by equation 4 is continuous, it is typically discretized by partitioning the time interval $[0, 1]$ into a finite sequence of time intervals $[t_n, t_{n+1})$. The QP problem is solved at each instant, and the system state is kept constant within each interval. This approximation affects the enforcement of consistency and safety constraints, and its impact depends on the time interval length. Taking CBF constraint as example, we give a detailed discussion below. Define the error of real-time CBF constraint and time instant CBF constraint as m, we have

$$\begin{aligned} m =& \frac{\mathrm{d}h}{\mathrm{d}x}\left(f(x,t) - \frac{1}{2}g(t)^2\nabla_x \log p_t^\theta(x)\right) - \frac{\mathrm{d}h}{\mathrm{d}x_n}\left(f(x_n,t_n) - \frac{1}{2}g(t_n)^2\nabla_{x_n} \log p_t^\theta(x_n)\right) \\ &+ \alpha(t)h(x) - \alpha(t_n)h(x_n) + \left(\frac{\mathrm{d}h}{\mathrm{d}x} - \frac{\mathrm{d}h}{\mathrm{d}x_n}\right)u_n, \end{aligned}$$

When the terms $f(x,t), g(t)^2\nabla_x \log p_t^\theta(x), h(x), \alpha(t)h(x)$ are locally Lipschitz, it follows that:

$$\|m\| \leq (c_1 + c_2 + c_3\|u_n\|)\|x - x_n\| \quad \forall t \in [t_n, t_{n+1})$$

where c_1, c_2, c_3 are the Lipschitz constants of $\frac{\mathrm{d}h}{\mathrm{d}x}\left(f(x,t) - \frac{1}{2}g(t)^2\nabla_x \log p_t^\theta(x)\right)$, $\alpha(t)h(x)$ and $\frac{\mathrm{d}h}{\mathrm{d}x}$, respectively. By closeness of solutions between x and x_n and the boundedness of u_n for $\|u_n\| \leq c_4$, it follows that:

$$\|m\| \leq \frac{(c_1 + c_2 + c_3c_4)c_5}{c_1 + c_2c_4}(e^{(c_1+c_2c_4)(t-t_n)} - 1) \quad \forall t \in [t_n, t_{n+1})$$

where $c_5 \geq \|f(x,t) - \frac{1}{2}g(t)^2\nabla_x \log p_t^\theta(x) + u_n\|$. Note that c_5 is guaranteed to exist over the interval $t \in [t_n, t_{n+1})$ due to the locally Lipschitz property of $f(x,t), g(t)^2\nabla_x \log p_t^\theta(x)$ and the boundedness of u_n.

From the upper bound of the constraint error m, it can be observed that when the sampling step size is excessively large, the value of $t - t_n$ increases as time t approaches the next discrete instant t_{n+1}. Therefore, the upper bound of the constraint error grows larger, which in turn degrades the ability of the safety constraints to guarantee system safety.

The above analysis also provides a direct way to compensate for the inter-sampling error. Let $\Delta t_n = t_{n+1} - t_n$ denote the sampling interval, and define the following upper bound evaluated at the end of the interval:

$$\bar{m}_{h,n} = \frac{(c_1 + c_2 + c_3c_4)c_5}{c_1 + c_2c_4}\left(e^{(c_1+c_2c_4)\Delta t_n} - 1\right).$$

Then, for every $t \in [t_n, t_{n+1})$, the CBF constraint error satisfies

$$|m(t)| \leq \bar{m}_{h,n}.$$

Therefore, instead of enforcing the nominal CBF condition only at the sampling instant, we can introduce a margin into the QP and impose

$$\frac{\mathrm{d}h}{\mathrm{d}x_n}\left(f(x_n,t_n) - \frac{1}{2}g(t_n)^2\nabla_{x_n}\log p_{t_n}^\theta(x_n) + u_n\right) + \alpha(t_n)h(x_n) \geq \bar{m}_{h,n}.$$

Since the continuous-time CBF expression can deviate from its sampling-instant value by at most $\bar{m}_{h,n}$, the above tightened constraint implies

$$\frac{\mathrm{d}h}{\mathrm{d}x}\left(f(x,t) - \frac{1}{2}g(t)^2\nabla_x \log p_t^\theta(x) + u_n\right) + \alpha(t)h(x) \geq 0,$$

for all $t \in [t_n, t_{n+1})$. Hence, the inter-sampling error can be explicitly compensated by tightening the CBF constraint according to the sampling interval and the corresponding Lipschitz bounds.

The same construction applies to the CLF condition. Let $\bar{m}_{V,n}$ denote an upper bound on the discretization error of the CLF expression over the interval $[t_n, t_{n+1})$. The CLF constraint can then be tightened as

$$\frac{\mathrm{d}V}{\mathrm{d}x_n}\left(f(x_n,t_n) - \frac{1}{2}g(t_n)^2\nabla_{x_n}\log p_{t_n}^\theta(x_n) + u_n\right) + \alpha(t_n)V(x_n) \leq -\bar{m}_{V,n} + \delta_n.$$

Consequently, the nominal discrete-time CLF-CBF constraints compensate for the worst-case inter-sampling errors. If the tightened QP remains feasible and $\delta_n \equiv 0$, the continuous-time CLF and CBF conditions hold throughout every sampling interval, thereby extending the consistency and safety guarantees to the discrete-time implementation.

The above compensation strategy requires valid upper bounds on the Lipschitz constants c_1, c_2, and c_3, as well as a bound on the guidance magnitude and the resulting state velocity. When valid Lipschitz bounds are available, the inter-sampling errors can be explicitly compensated by tightening the CLF and CBF constraints. This provides a direct route to extending the continuous-time guarantees to the sampled-data implementation. However, obtaining tight and uniform Lipschitz bounds for the score-based diffusion dynamics and the guidance term can be difficult in practice.

Importantly, the error analysis above shows that the discretization error vanishes as the sampling interval decreases. In particular, for a locally Lipschitz system, the bound satisfies

$$\bar{m}_{h,n}, \bar{m}_{V,n} = \mathcal{O}(\Delta t_n) \quad \text{as} \quad \Delta t_n \to 0.$$

This observation motivates our progressive distillation strategy. Rather than directly applying a large sampling interval, progressive distillation gradually reduces the sampling horizon and transfers the continuous-time CLF-CBF properties across intermediate sampling schedules. Consequently, each distillation stage operates in a regime with a smaller inter-sampling error, while the final student model benefits from a substantially reduced sampling cost.

Thus, progressive distillation is not merely an empirical acceleration technique. It provides a discretization-aware mechanism that progressively reduces the gap between the continuous-time guided dynamics and their finite-step implementation. When Lipschitz bounds are available, the remaining error can be explicitly compensated through constraint tightening; when these bounds are unknown, progressive distillation still guarantees that the discretization error decreases asymptotically with the sampling interval under the local Lipschitz assumption.

C More details about experiments

- **Umaze:** A point-agent must navigate from a specified start to a goal position in obstacle-filled mazes. The planned trajectory must avoid collisions with obstacles to satisfy safety requirements. Training trajectories were generated by a classical navigation planner tailored to each maze configuration.
- **Hopper and Walk2d:** These tasks require a one-legged (Hopper) and a bipedal (Walk2d) robot, respectively, to move forward via jumping or walking. Safety is enforced by constraining the robot's torso center height to remain below a prescribed threshold. The training data is publicly available in Janner et al. (2022).
- **Manipulation:** The task requires generating joint-space trajectories that drive a robot to grasp and place objects at specified locations. Safety is imposed through joint limits to avoid self-collision and kinematic violations in joint space. Training trajectories were produced using the PDDLStream planner Garrett et al. (2020).

C.1 Safety constraints setting for benchmark task

Umaze: The objective is to generate a motion trajectory that drives the agent from its initial position to a randomly assigned target position, while avoiding collisions with obstacles within the maze. The state dimension of trajectory is 2. The problem includes two obstacle constraints. The first constraint, defined by a super-ellipsoid property, is given by

$$\left(\frac{x-x_0}{a_1}\right)^2+\left(\frac{y-y_0}{b_1}\right)^2\geq 1,$$

where (x, y) is the position of agent, (x_0, y_0) is the location of the inner obstacle, a_1, b_1 are the width and height of obstacle. The second constraint is for the entire maze boundary, which is also formulated as a super-ellipse:

$$\left(\frac{x-x_1}{a_2}\right)^2+\left(\frac{y-y_1}{b_2}\right)^2\leq 1,$$

where (x_1, y_1) is the location of the maze, a_2, b_2 are the width and height of maze. The above constraint both are normalized according to the normalization of (x, y) along the x-axis and y-axis.

Locomotion: The robot is required to move forward while keeping its height below a predefined threshold h_r. The trajectory state dimension is 3 for the Hopper robot and 6 for the Walker2d robot. Accordingly, the robot's motion height z must satisfy the following safety constraint:

$$z+\psi v_z\leq h_r$$

where $\psi > 0$ and v_z is the speed of robot along the z-axis. The safety constraint is more robust than a simple height constraint $z \leq h_r$, as it enforces a larger safety distance when the robot performs fast jumps. The constraint also requires normalization.

Manipulation: The manipulator performs grasping and placing tasks, where the diffusion model generates joint trajectories for the manipulator based on the positions of the objects to be grasped

and placed. The state dimension of trajectory is 7. The safety constraint is defined as the joint angle limits of the manipulator:

$$x_{min} \leq x + \psi v \leq x_{max}$$

where x_{min}, x_{max} are the minimum and maximum joint limits, respectively. v is the joint speed corresponding to the joint angle x, which is calculated by x_n and x_{n+1} along the planning horizon. The joints limits are also normalized as in the simple specification case.

Table 6 compares progressive distillation with CLF-only and CBF-only guidance under different sampling budgets. The CLF-only variant generally achieves lower ATE and FTE, indicating that the CLF effectively guides the student model toward the teacher output and preserves teacher–student consistency. However, its safety values are smaller and may become negative as the number of sampling steps decreases, showing that consistency guidance alone cannot prevent safety violations.

In contrast, the CBF-only variant generally provides higher safety function values and more robust safety behavior under aggressive sampling reduction. However, removing the CLF guidance leads to larger ATE and FTE, especially at fewer sampling steps, indicating a larger deviation from the teacher trajectory. The inference times of the two variants are comparable, which suggests that the main difference between them arises from the guidance objectives rather than additional computational overhead. These results demonstrate that CLF and CBF guidance play complementary roles: the CLF preserves consistency with the teacher model, whereas the CBF enforces safety. Therefore, combining both constraints, as in SPDD, is necessary to achieve a favorable balance between trajectory consistency and safety.

Table 6: Ablation study of CLF-only and CBF-only guidance in progressive distillation across different tasks and sampling budgets.

Experiment	Metric	PD-CLF-only			PD-CBF-only		
		320	160	80	320	160	80
Maze	ATE	1.32	1.54	1.70	1.53	1.58	1.70
	FTE	0.08	0.08	0.07	0.23	0.34	0.50
	Inf. time	70.45	61.73	50.48	70.43	60.92	49.26
	safety	0.05	0.03	0.02	0.14	0.15	0.14
Hopper	ATE	0.14	0.20	0.23	0.25	0.33	0.43
	FTE	0.05	0.07	0.06	0.11	0.16	0.19
	Inf. time	22.58	19.64	15.81	23.03	18.92	15.43
	safety	−0.00	−0.00	−0.00	0.01	0.00	0.00
Walk2d	ATE	0.17	0.22	0.25	0.30	0.41	0.48
	FTE	0.08	0.07	0.07	0.09	0.11	0.13
	Inf. time	34.28	29.85	24.35	35.08	30.27	21.79
	safety	0.00	−0.01	−0.01	0.01	0.01	0.00
Manipulation	ATE	0.24	0.30	0.30	0.39	0.42	0.51
	FTE	0.07	0.08	0.09	0.18	0.26	0.29
	Inf. time	41.00	30.02	19.24	39.25	28.42	19.67
	safety	−0.01	−0.00	−0.00	0.01	0.01	0.01

C.2 Training details

All diffusion models in the experiments are trained and evaluated on an NVIDIA RTX 3090 GPU with 16 GB memory. The detailed values of the training hyperparameters are listed in Table 7.

C.3 Deployment on real-world robotic platforms

Although our current implementation mainly focuses on simulation environments, the modular design of SPDD makes it compatible with standard robotic deployment pipelines. The proposed framework can be integrated with onboard perception and control systems, while the safety constraints (e.g., actuator limits and workspace boundaries) can be adapted according to real-world specifications. By

Table 7: Training hyperparameters for all methods across different tasks.

SafeDiffuser	Umaze	Hopper	Walker2d	Manipulation
Batch size	32	32	32	32
Horizon	384	40	40	120
Learning rate	$2e^{-4}$	$2e^{-4}$	$2e^{-4}$	$2e^{-4}$
Steps	$1e^{6}$	$1e^{6}$	$1e^{6}$	$7e^{5}$
CD	**Umaze**	**Hopper**	**Walker2d**	**Manipulation**
Batch size	32	32	32	32
Horizon	384	40	40	120
Learning rate	$2e^{-4}$	$2e^{-4}$	$2e^{-4}$	$2e^{-4}$
Steps	$1e^{6}$	$1e^{6}$	$1e^{6}$	$7e^{5}$
SPDD	**Umaze**	**Hopper**	**Walker2d**	**Manipulation**
Batch size	32	32	32	32
Horizon	384	40	40	120
Learning rate	$2e^{-4}$	$2e^{-4}$	$2e^{-4}$	$2e^{-4}$
Steps	$1e^{6}$	$1e^{6}$	$1e^{6}$	$7e^{5}$

significantly reducing the sampling steps, SPDD is also suitable for resource-constrained robotic platforms and real-time control settings.

Overall, SPDD introduces only moderate computational overhead compared to existing diffusion-based planners, while providing formal guarantees on safety and consistency. Given the demonstrated deployment capability of related diffusion-based frameworks and the compatibility of our method with efficient sampling and standard control pipelines, we believe SPDD can be integrated into real-world robotic systems with limited adaptation. Exploring large-scale real-world deployment remains an important direction for future work.

C.4 Broader impacts

Potential Positive Societal Impacts. The proposed method can enable faster and safer trajectory generation for robotic systems, which is particularly beneficial in safety-critical applications such as autonomous driving, healthcare robotics, and industrial automation. By improving inference efficiency, it lowers the computational barrier for deploying advanced planning algorithms on resource-constrained platforms, potentially broadening access to intelligent robotic technologies. The integration of formal safety constraints (e.g., CLF-CBF) also promotes more reliable and predictable system behavior, reducing the risk of accidents in real-world deployments.

Potential Negative Societal Impacts. Despite these benefits, the method could be misused in autonomous systems where safety guarantees are assumed but not fully satisfied under unmodeled conditions, potentially leading to harmful outcomes. The reliance on predefined safety constraints may result in a false sense of security if these constraints do not capture all real-world risks. Additionally, improved efficiency in trajectory generation could accelerate the deployment of autonomous systems in sensitive domains (e.g., surveillance or military applications), raising ethical concerns. Finally, failures due to model mismatch, dynamic environments may still occur, which could have negative consequences if not properly monitored or mitigated.