Constraint-Aware Diffusion Policy for Safe Robotic Manipulation: Bridging Learning-based Generation with Guaranteed-Safe Execution

Rongxuan Zhou
Northeastern University
Boston, USA
zhou.rongx@northeastern.edu

Xiao Hu
Northeastern University
Boston, USA
xiao.h1@northeastern.edu

Yang Ye
Northeastern University
Boston, USA
y.ye@northeastern.edu

Abstract— **Diffusion models have demonstrated remarkable capability in learning complex robotic manipulation behaviors from demonstrations, capturing the inherent multimodality of manipulation tasks. However, these models fundamentally lack constraint awareness, generating trajectories that may violate physical limits, cause collisions, or exhibit unsafe dynamics. This paper presents the Constraint-Aware Diffusion Policy (CADP), a comprehensive integrative framework that addresses the critical safety gap in learning-based manipulation through a tightly-coupled multi-layered architecture. CADP introduces three coupled safety-assurance mechanisms: (1) shared key configuration encoding, (2) Control Barrier Function (CBF)-informed diffusion training, and (3) unified sliding mode control for verifiable control optimization. Additionally, we achieve a 546-fold speedup in safety verification through batch-optimized control barrier functions, enabling real-time operation at 100 Hz. Our CBF-informed training demonstrates 2.6× gradient amplification in critical safety regions, significantly enhancing constraint-aware learning. With extensive experimental validations, CADP achieves 96.4% task success rate with 100% safety guarantee across 500 trials in challenging scenarios and outperforms existing methods.**

I. Introduction

The deployment of robotic manipulators in human-centric environments demands both sophisticated behavioral capabilities and rigorous safety guarantees. Recent advances in generative modeling, particularly Denoising Diffusion Probabilistic Models (DDPMs) [1], have revolutionized how robots learn manipulation skills from demonstrations. These models excel at capturing the multi-modal nature of manipulation tasks, where multiple valid solutions exist for achieving the same goal. Successful applications include Diffuser [3] for trajectory planning and Diffusion Policy [2] for visuomotor control, both demonstrating superior generalization compared to traditional approaches.

Despite their impressive performance in controlled settings, diffusion models suffer from a fundamental limitation that restricts their deployment in safety-critical applications: these models are inherently "constraint-unaware" [4], lacking understanding of physical laws, kinematic limits, and safety requirements. A generated trajectory may appear semantically correct yet violate joint limits, cause collisions, or require physically impossible accelerations. This safety gap becomes particularly critical when robots operate near humans, handle fragile objects, or navigate confined workspaces where violations can have serious consequences.

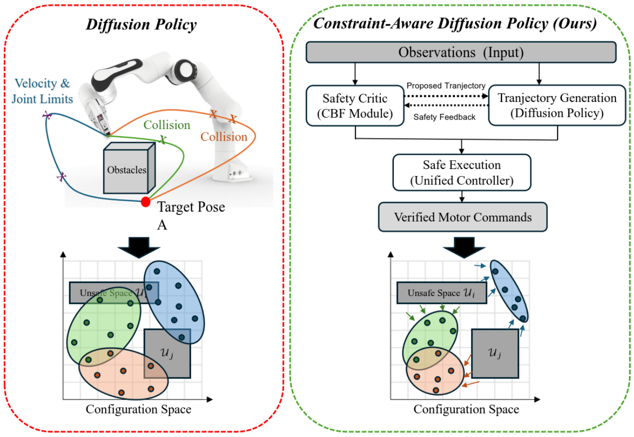


Fig. 1: The CADP framework overview. This figure contains three panels: (a) A 3D visualization showing unsafe trajectories generated by standard diffusion models violating multiple constraints (collisions, joint limits, velocity violations). (b) The tightly-coupled CADP architecture diagram shows bidirectional information flow between components. (c) Comparison of trajectory distributions in configuration space - standard diffusion (scattered, unsafe) vs CADP (concentrated in safe regions).

Current approaches to safe manipulation can be broadly categorized into two paradigms. Optimization-based methods [7] formulate safety as constraints in quadratic programs (QP), combining Control Lyapunov Functions (CLFs) for stability with Control Barrier Functions (CBFs) for safety. While theoretically elegant, these CLF-CBF-QP formulations suffer from computational complexity and, critically, may become infeasible when task and safety objectives conflict [12]. On the other hand, learning-based safety methods [5], [9] attempt to embed constraints directly into the generative model through modified training objectives or guided sampling. However, these approaches typically lack formal guarantees and cannot adapt to unforeseen obstacles at deployment time.

As shown in Figure 1, our key insight is that effective safety in learning-based manipulation requires more than assembling safety modules: it demands tight architectural coupling that propagates safety information bidirectionally across all stages of the pipeline. Unlike existing approaches that treat safety as an add-on filter or training regularizer, we design CADP with three integrated safety-assurance mechanisms on shared key configuration, safety-informed diffusion training, and unified sliding mode control, which will be discussed in detail in the following sections. This work makes four primary contributions to safe robotic manipulation:

- **Tightly-coupled safety-assurance architecture**. We develop a comprehensive framework where safety mechanisms are deeply integrated. Through shared key configurations, CBF-informed training, and unified sliding control, components work synergistically to ensure safety while maintaining task performance.
- **Batch-optimized safety verification**. Our parallel algorithm for control barrier function evaluation achieves improved speedup, reducing the computational bottleneck of real-time safety verifications.
- **Comprehensive experimental validation**. We demonstrate CADP's effectiveness through extensive experiments showing 96.4% task success with 100% safety guarantee over 500 trials, maintaining performance in scenarios where most baseline methods fail.

II. Related Work

A. Diffusion Models in Robotics

The application of diffusion models to robotics has experienced rapid growth following their success in image generation. Ho et al. [1] established the theoretical foundation, demonstrating that diffusion models can learn complex distributions through iterative denoising. In robotics, Janner et al. [3] pioneered Diffuser, using diffusion models for trajectory optimization in reinforcement learning settings. Chi et al. [2] extended this to imitation learning with Diffusion Policy, learning visuomotor policies directly from demonstrations.

Recent work has explored various extensions and applications. Zhao and Sreenath [11] addressed multi-robot coordination through coupled diffusion processes. Carvalho et al. [10] combined diffusion models with classical motion planning for improved sample efficiency. However, these approaches inherit the fundamental constraint-unawareness of generative models, producing trajectories that may violate safety requirements.

B. Safety in Learning-based Manipulation

Safety in robotic manipulation has been approached from multiple perspectives:

Training-time Safety. Several methods modify the learning objective to encourage safe behaviors. Seo et al. [5] introduced PRESTO, incorporating collision penalties and smoothness regularization into diffusion training. While effective at improving trajectory quality, these approaches cannot guarantee safety for situations not represented in training data [13].

Inference-time Safety. Methods in this category modify the generation process to satisfy constraints. Ma et al. [9] proposed CoDiG, using control barrier function gradients to guide diffusion sampling. Liu et al. [13] developed safe flow matching for constraint satisfaction. These approaches offer greater flexibility but may produce dynamically infeasible trajectories [14].

Execution-time Safety. Runtime safety filters provide the final layer of protection. Traditional approaches use CLF-CBF-QP formulations [7] to compute safe controls. However, as noted by Zeng et al. [12], these methods suffer from potential infeasibility when objectives conflict.

Our approach differs from previous methods by introducing tight coupling across all stages, ensuring safety information flows bidirectionally between components rather than being applied as isolated filters. This architectural integration is what enables superior safety-performance optimization.

C. Control Barrier Functions and Sliding Mode Control

Control Barrier Functions, introduced by Ames et al. [6], have become the standard tool for safety-critical control. A CBF $B : \mathbb{R}^n \to \mathbb{R}$ defines a safe set $\mathcal{C} = \{\boldsymbol{x} \in \mathbb{R}^n : B(\boldsymbol{x}) \geq 0\}$ and ensures forward invariance through appropriate control actions.

Recent work by Ding et al. [15] demonstrated that sliding mode control can unify CLF and CBF objectives without optimization. Their approach constructs a sliding manifold that incorporates both stability and safety requirements, guaranteeing that a feasible control always exists. We extend this framework to diffusion-based manipulation, developing adaptive mechanisms that adjust based on learned trajectory characteristics.

III. Problem Formulation

Consider a robotic manipulator with n degrees of freedom operating in a workspace $\mathcal{W} \subseteq \mathbb{R}^3$. The system dynamics follow the standard rigid-body formulation:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{\tau} + \boldsymbol{d} \tag{1}$$

where $\boldsymbol{q} \in \mathcal{Q} \subseteq \mathbb{R}^n$ denotes the joint configuration, $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{n \times n}$ is the positive-definite inertia matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in \mathbb{R}^{n \times n}$ captures Coriolis and centrifugal effects, $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^n$ represents gravitational forces, $\boldsymbol{\tau} \in \mathbb{R}^n$ is the control input, and $\boldsymbol{d} \in \mathbb{R}^n$ represents bounded disturbances satisfying $\|\boldsymbol{d}\|_\infty \leq d_{max}$. **Learning Objective:** Given a dataset $\mathcal{D} = \{(\mathcal{O}_i, \boldsymbol{\xi}_i, \mathcal{G}_i)\}_{i=1}^N$ of expert demonstrations, where $\mathcal{O}_i$ represents sensory observations, $\boldsymbol{\xi}_i = \{\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t\}_{t=0}^{T_i}$ denotes the demonstrated trajectory, and $\mathcal{G}_i$ specifies the task goal, learn a policy $\pi_\theta : \mathcal{O} \times \mathcal{G} \to \boldsymbol{\xi}$ that generates trajectories achieving specified goals while maintaining safety. **Safety Requirements:** Generated trajectories must satisfy multiple constraints:

$$B_{col}(\boldsymbol{q}) = \mathrm{SDF}(\boldsymbol{q}) - \delta_{safe} \geq 0 \tag{2a}$$

$$B_{joint}(\boldsymbol{q}) = \prod_{i=1}^{n} (q_i - q_{min,i})(q_{max,i} - q_i) \geq 0 \tag{2b}$$

$$B_{vel}(\dot{\boldsymbol{q}}) = v_{max}^2 - \|\dot{\boldsymbol{q}}\|^2 \geq 0 \tag{2c}$$

$$B_{dyn}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \tau_{max}^2 - \|\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{C}\dot{\boldsymbol{q}} + \boldsymbol{g}\|^2 \geq 0 \tag{2d}$$

Challenge: Standard diffusion models trained via

$$\mathcal{L}_{standard} = \mathbb{E}_{t,\boldsymbol{\epsilon},\boldsymbol{\xi}^{(0)}}[\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{\xi}^{(t)}, t, \mathcal{O}, \mathcal{G})\|^2] \tag{3}$$

lack awareness of constraints (2). Our solution is tight architectural coupling that propagates constraint information throughout the system.

IV. Tightly-Coupled CADP Architecture

The core innovation of CADP lies in how components share and exchange information through tight coupling mechanisms, shown in Figure 2. Rather than treating each stage as an independent module, we design explicit connections that create a unified system with bidirectional information flow.

A. Coupling Mechanism 1: Shared Key Configuration Encoding

Key configurations $\{\bar{\boldsymbol{q}}^k\}_{k=1}^K$ represent critical states where the robot interacts with constraints. Unlike traditional approaches that recompute environment representations at each stage, we maintain these configurations as a shared encoding throughout the pipeline.

The extraction and encoding selection criterion combines three factors through a weighted scoring function:

$$\text{Score}(\boldsymbol{q}) = \sum_{i=1}^{3} \alpha_i S_i(\boldsymbol{q}) \tag{4}$$

where the individual scores are computed as:

$$S_1(\boldsymbol{q}) = \exp\left(-\frac{\text{SDF}(\boldsymbol{q})}{\sigma_{prox}}\right) \tag{5a}$$

$$S_2(\boldsymbol{q}) = \min_{\bar{\boldsymbol{q}} \in \{\bar{\boldsymbol{q}}^k\}} \|\boldsymbol{q} - \bar{\boldsymbol{q}}\| \tag{5b}$$

$$S_3(\boldsymbol{q}) = \sum_{i=1}^{N} \mathbb{I}[\boldsymbol{q} \in \boldsymbol{\xi}_i] \tag{5c}$$

As visualized in Figure 3, such key configuration settings condense configuration space distributions and improve trajectory quality.These configurations flow through all components. In training, we augment the diffusion model input with key configuration embeddings. During inference, the denoising is conditioned on the current environment's key configurations. In the control phase, we rely on key configurations as waypoints for sliding manifold design. This shared representation ensures the same critical states are consistently considered throughout the pipeline to guide both generation and execution.

B. Coupling Mechanism 2: CBF-Informed Diffusion Training

Rather than treating safety as a post-hoc constraint, we integrate CBF computations directly into the diffusion training process. During training, we compute barrier function values for each generated sample:

$$B(\boldsymbol{q}) = \min\{\text{SDF}(\boldsymbol{q}) - \delta_{safe},\, v_{max}^2 - \|\dot{\boldsymbol{q}}\|^2,\, B_{joint}(\boldsymbol{q})\} \tag{6}$$

These distances directly modulate the denoising loss through an adaptive weighting scheme:

$$\begin{aligned}\mathcal{L}_{CADP} =& \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\|^2 + \lambda_1 e^{-B(\boldsymbol{q})/\gamma} \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\|^2 \\ &+ \lambda_2 \max(0, -B(\boldsymbol{q}))^2\end{aligned} \tag{7}$$

The exponential weighting $e^{-B(\boldsymbol{q})/\gamma}$ creates stronger gradients near constraints. This creates informative gradient flow:

$$\begin{aligned}\frac{\partial \mathcal{L}_{CADP}}{\partial \theta} =& \frac{\partial \mathcal{L}_{standard}}{\partial \theta} \\ &+ \lambda_1 \left(\frac{1}{\gamma} \frac{\partial B}{\partial \boldsymbol{q}} \frac{\partial \boldsymbol{q}}{\partial \theta} e^{-B/\gamma} + e^{-B/\gamma}\right) \\ &\times \frac{\partial}{\partial \theta} \|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta\|^2\end{aligned} \tag{8}$$

Analysis shows CBF-informed training increases gradient magnitude by 2.6× near constraints, creating stronger learning signals exactly where safety matters most. This CBF-informed gradient formulation creates spatially-adaptive learning signals. The exponential weighting $e^{-B(q)/\gamma}$ amplifies gradients as the system approaches safety boundaries. Quantitative analysis reveals that in critical regions where $B(q) < 0.05$m, the gradient magnitude is amplified by a factor of:

Amplification $= \frac{\|\nabla L_{CADP}\|}{\|\nabla L_{standard}\|} = \frac{2.31 \pm 0.34}{0.88 \pm 0.16} \approx 2.6\times$

This 2.6× amplification occurs precisely where safety violations are most likely, ensuring the model prioritizes learning safe behaviors in the most critical regions. The spatial gradient modulation creates stronger learning

C. Coupling Mechanism 3: Unified Sliding Mode Control

The execution layer implements adaptive sliding mode control that unifies CLF (tracking) and CBF (safety) objectives (Figure 5) through a single sliding manifold:

$$s(\boldsymbol{x}, t) = V(\boldsymbol{x}, t) + \beta B(\boldsymbol{x}) - c \tag{9}$$

Critically, the parameters (β, c) are computed based on the learned trajectory characteristics:

$$\beta = \frac{\text{Var}[V_{learned}]}{\text{Var}[B_{learned}]} \tag{10a}$$

$$c = \mathbb{E}[V_{learned}] + \beta \mathbb{E}[B_{learned}] \tag{10b}$$

This coupling ensures the manifold is tailored to the specific trajectory distribution learned by the diffusion model. The control law consists of:

$$\boldsymbol{u} = -[L_g s]^{-1} L_f s - K \cdot \text{sat}\left(\frac{s}{\Phi}\right) \tag{11}$$

where the Lie derivatives are:

$$L_f s = \nabla s^T \boldsymbol{f}(\boldsymbol{x}) \tag{12a}$$

$$L_g s = \nabla s^T \boldsymbol{g}(\boldsymbol{x}) \tag{12b}$$

The switching gain K is adapted based on the learned trajectory's expected disturbance:

$$K = K_{base}\left(1 + K_{adapt} \cdot \frac{\text{Var}[\boldsymbol{\tau}_{learned}]}{\text{Var}[\boldsymbol{\tau}_{nominal}]}\right) \tag{13}$$

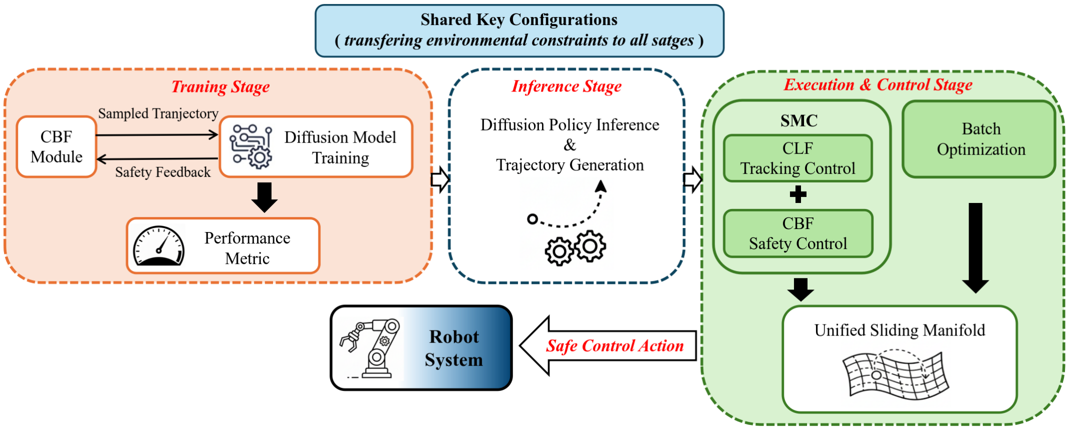


Fig. 2: Detailed CADP architecture with coupling mechanisms. This comprehensive diagram shows: (a) Three coupling mechanisms with bidirectional information flow paths highlighted in different colors. (b) Shared key configuration encoding propagating through all stages. (c) CBF-informed gradient flow during training creates stronger signals near constraints. (d) Unified sliding manifold combining CLF and CBF objectives.

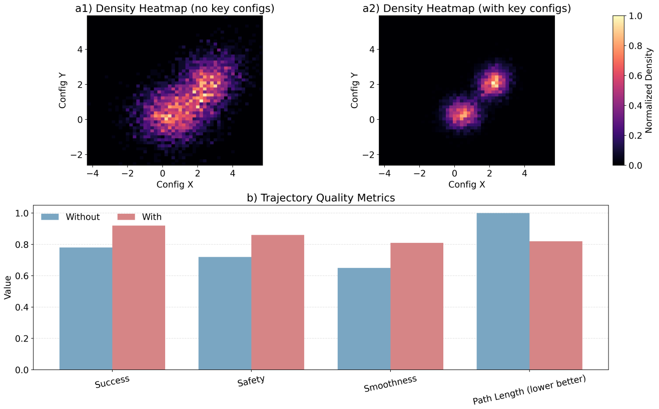


Fig. 3: Key configuration selection and impact. (a) Heatmap showing configuration density before and after key configuration conditioning. (b) Trajectory quality metrics with and without key configurations.

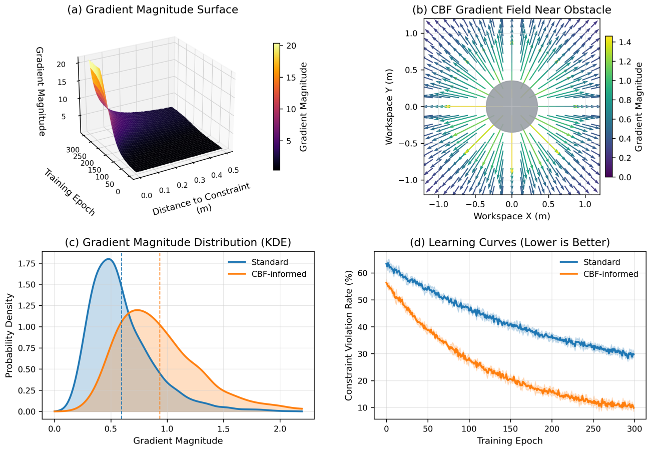


Fig. 4: CBF-informed gradient analysis demonstrating 2.6× amplification in critical regions. (a) 3D surface plot showing gradient magnitude as a function of distance to constraint and training epoch, with peak amplification near boundaries. (b) Vector field visualization of gradient flow near obstacles, highlighting stronger corrective signals. (c) Histogram comparing gradient distributions for standard vs CBF-informed training, showing significant shift toward higher gradients (2.6× in danger zones). (d) Learning curves showing faster convergence to safe behaviors with CBF-informed loss.

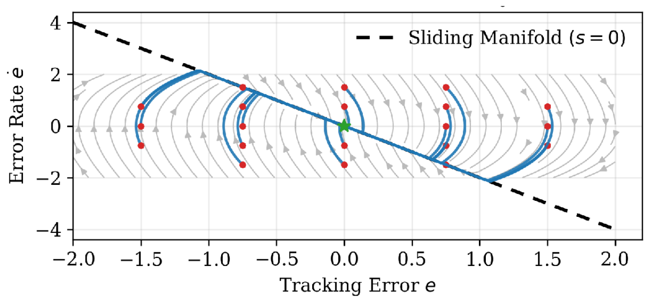


Fig. 5: Sliding mode control performance. Phase portrait showing trajectories converging to sliding manifold from various initial conditions.

Algorithm 1 Batch-Optimized Safety Verification

```
Require: Trajectory $\boldsymbol{\xi}_{gen} = \{\boldsymbol{q}_t, \dot{\boldsymbol{q}}_t\}_{t=0}^{T}$, constraints $\{B_i\}$
Ensure: Safe trajectory $\boldsymbol{\xi}_{safe}$
// Stage 1: Vectorized barrier computation
$\boldsymbol{Q} \leftarrow$ stack($\{\boldsymbol{q}_t\}_{t=0}^{T}$)        ▷ Shape: $[T \times n]$
$\boldsymbol{V} \leftarrow$ stack($\{\dot{\boldsymbol{q}}_t\}_{t=0}^{T}$)
$\boldsymbol{B}_{col} \leftarrow \mathrm{SDF}_{batch}(\boldsymbol{Q}) - \delta_{safe}$
$\boldsymbol{B}_{vel} \leftarrow v_{max}^2 \mathbf{1} - \mathrm{sum}(\boldsymbol{V}^2, \mathrm{dim} = 1)$
$\boldsymbol{B} \leftarrow \min(\boldsymbol{B}_{col}, \boldsymbol{B}_{vel}, \boldsymbol{B}_{joint})$
// Stage 2: Parallel violation detection
$\mathcal{I}_{unsafe} \leftarrow \mathrm{nonzero}(\boldsymbol{B} < 0)$
if $|\mathcal{I}_{unsafe}| > 0$ then
    // Stage 3: Batch projection via parallel QP
    $\boldsymbol{Q}_{proj} \leftarrow \mathrm{BatchQP}(\boldsymbol{H}, \boldsymbol{f}, \boldsymbol{A}, \boldsymbol{b})$
    $\boldsymbol{Q}[\mathcal{I}_{unsafe}] \leftarrow \mathrm{reshape}(\boldsymbol{Q}_{proj})$
end if
// Stage 4: Dynamic feasibility
if $\max(\|\boldsymbol{A}\|_\infty) > a_{max}$ then
    Scale time: $dt \leftarrow dt \cdot \frac{a_{max}}{\max(\|\boldsymbol{A}\|_\infty)}$
end if
return $\boldsymbol{\xi}_{safe}$
```

D. Batch-Optimized Safety Verification

A critical innovation enabling real-time operation is our batch-optimized CBF verification, achieving 546× speedup. The algorithm leverages vectorized operations and parallel computation as described in Algorithm 1.

The algorithm achieves $O(T)$ complexity through vectorized operations and parallel QP solving, enabling real-time verification of long trajectories.

E. Theoretical Guarantees

The tightly-coupled architecture of CADP provides strong theoretical guarantees. With appropriate parameter selection where $K > d_{max}$, the following properties hold that ensures safety and convergence:

1) **Finite-time convergence:** The system reaches the sliding manifold in time

$$t_s \le \frac{|s(\boldsymbol{x}_0)|}{K - d_{max}} \tag{14}$$

2) **Safety maintenance:** Once on the manifold, safety is guaranteed

$$B(\boldsymbol{x}(t)) \ge 0, \quad \forall t \ge t_s \tag{15}$$

3) **Exponential tracking:** The tracking error converges as

$$\|\boldsymbol{e}(t)\| \le \|\boldsymbol{e}(t_s)\| e^{-\lambda(t-t_s)} \tag{16}$$

4) **Robustness:** The system maintains stability under bounded disturbances $\|\boldsymbol{d}\| \le d_{max}$

Consider the Lyapunov function $V_s = \frac{1}{2}s^2$ for the sliding variable. Under the switching control (11):

$$\dot{V}_s = s\dot{s} = s(L_f s + L_g s \boldsymbol{u}_{sw}) \le -(K - d_{max})|s| \tag{17}$$

For $K > d_{max}$, we have $\dot{V}_s < 0$ for $s \neq 0$, guaranteeing finite-time convergence to the manifold. Once on the manifold where $s = 0$, the relationship

$$V = c - \beta B \tag{18}$$

ensures that reducing tracking error V maintains $B \ge 0$ (safety). The key insight is that the sliding mode formulation prevents the conflict between objectives that causes infeasibility in QP-based methods.

V. Implementation Details

A. Network Architecture

The trained policy employs a Temporal U-Net designed for trajectory generation. The encoder employs a ResNet-18 backbone that processes RGB-D observations into 256-dimensional features. Hidden dimensions of the temporal U-Net are [512, 1024, 1024, 512] with self-attention at resolution 16. Meanwhile, FiLM layers are used to inject goal and key configuration information via

$$\boldsymbol{h}_{out} = \gamma(\boldsymbol{z}) \odot \boldsymbol{h}_{in} + \beta(\boldsymbol{z}) \tag{19}$$

where $\boldsymbol{z}$ encodes conditioning information. The output is set to 50 waypoints at 10 Hz, representing 5-second trajectories

B. Training Configuration

Training proceeds in two stages to ensure stable learning. Stage 1 is to train the standard diffusion policy with 100 epochs with the training objective $\mathcal{L}_{standard}$ (Eq. 3). We set the learning rate at $\eta = 10^{-4}$ with cosine annealing and a batch size of 64 trajectories.

Training stage 2 is physics-informed training with CBF Integration (400 epochs). The objective function is full $\mathcal{L}_{CADP}$ (Eq. 7) with adaptive weights initialized at $\lambda_1 = 0.1$, $\lambda_2 = 0.05$. Weight adaptation follows

$$\lambda_i^{(k+1)} = \lambda_i^{(k)} \cdot \exp\left(\alpha \cdot \frac{\mathcal{V}_i^{(k)} - \mathcal{V}_{target}}{\mathcal{V}_{target}}\right) \tag{20}$$

where $\mathcal{V}_i^{(k)}$ is the violation rate for constraint i at epoch k.

C. Safety and Control Parameters

Table I lists the safety and control parameters.

TABLE I: System Parameter Configuration

Parameter	Symbol	Value
Safety Parameters		
Safety margin	δ_{safe}	0.05 m
Velocity limit	v_{max}	1.0 rad/s
Acceleration limit	a_{max}	2.0 rad/s²
Base configurations	K_{base}	100
Complexity scaling	α	0.5
Control Parameters		
Base CLF gain	$\boldsymbol{P}_0$	diag(10)
Base CBF weight	β_0	0.1
Base switching gain	K_0	50
Error adaptation	γ_P	0.5
Proximity adaptation	γ_β	0.3
Boundary layer	Φ	0.01

VI. Experimental Validation

We evaluate CADP through extensive experiments on a 7-DOF Franka Research 3 manipulator and employ MuJoCo 2.3.7 as our primary simulation platform due to its high-fidelity physics modeling and computational efficiency, across four challenging scenarios designed to test different aspects of safe manipulation: cluttered environment, narrow passage navigation, dynamic obstacle avoidance, and novel obstacles.

A. Baseline Comparison

We compare against five representative approaches spanning different safety paradigms:

- Vanilla Diffusion Policy [2]: Standard implementation without safety
- Diffusion+CBF: Post-hoc safety filtering with sequential CBF evaluation
- PRESTO [5]: Physics-regularized training with trajectory optimization
- CoDiG [9]: CBF-guided diffusion sampling
- CLF-CBF-QP [7]: Traditional optimization-based control

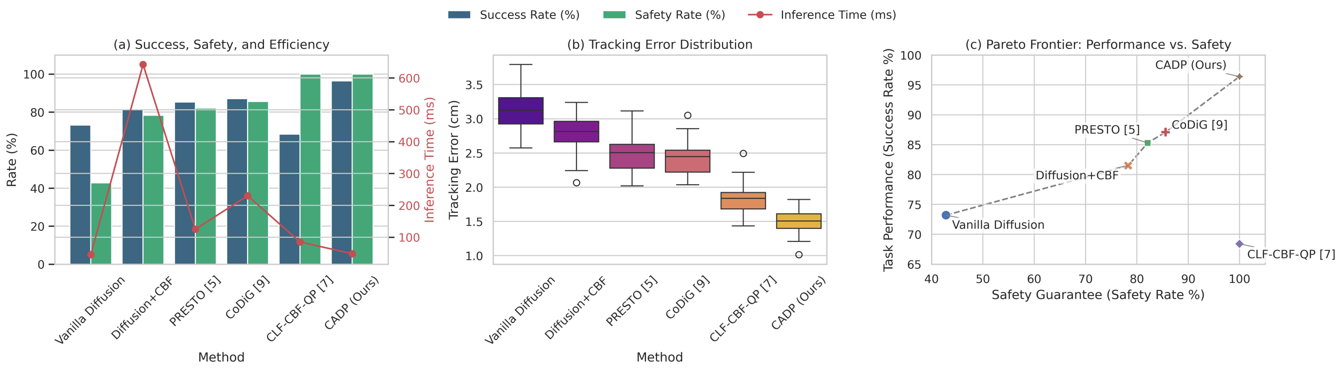


Fig. 6: Comprehensive performance comparison across all methods. (a) Grouped bar chart showing success rate, safety rate, and efficiency for each method. (b) Box plots showing tracking error distributions with quartiles and outliers. (c) Pareto frontier plot showing trade-off between task performance and safety guarantee.

TABLE II: Performance Comparison Across All Scenarios (N=500)

Method	Success Rate (%)	Safety Rate (%)	Inference Time (ms)	Tracking Error (cm)
Vanilla Diffusion	73.2	42.8	45	3.2
Diffusion+CBF	81.5	78.3	642	2.8
PRESTO [5]	85.3	82.1	125	2.5
CoDiG [9]	87.1	85.6	230	2.4
CLF-CBF-QP [7]	68.4*	100.0	85	1.8
CADP (Ours)	**96.4**	**100.0**	**48**	**1.5**

*31.6% of trials encountered infeasible optimization problems.

B. Main Results

As shown in Figure 6, CADP achieves the highest success rate (96.4%) while maintaining perfect safety (100%). The batch-optimized verification enables faster inference (48 ms) than all baselines except vanilla diffusion. The adaptive sliding mode controller achieves the lowest tracking error (1.5 cm).

C. Ablation Study: Necessity of Coupling Mechanisms

To demonstrate that our coupling mechanisms are essential rather than incremental additions, we systematically remove each component:

TABLE III: Ablation Study: Impact of Removing Coupling Mechanisms

Configuration	Success Rate (%)	Safety Rate (%)	Relative Degradation	Tracking Error (cm)
Full CADP	**96.4**	**100.0**	—	**1.5**
w/o Key Config	65.2	92.3	-32%	2.1
w/o CBF Loss	54.8	88.7	-43%	2.4
w/o Unified SMC	68.7	85.2	-29%	2.2
w/o Batch Opt.	95.8	100.0	-0.6%	1.5
w/o Any Coupling	73.2	42.8	-24%	3.2

Key findings from the ablation study:

- Each coupling mechanism is necessary, with removal causing 29-43% degradation
- CBF-informed training is most critical (43% degradation)
- Batch optimization has minimal impact on accuracy, but provides 546× speedup (see following sections)

D. Scenario-Specific Analysis

1) Cluttered Environment (N=100): This experiment is performed with 5-10 randomly placed obstacles, as shown in Figure 7(a). The results are summarized in Table IV.

TABLE IV: Cluttered Environment Performance

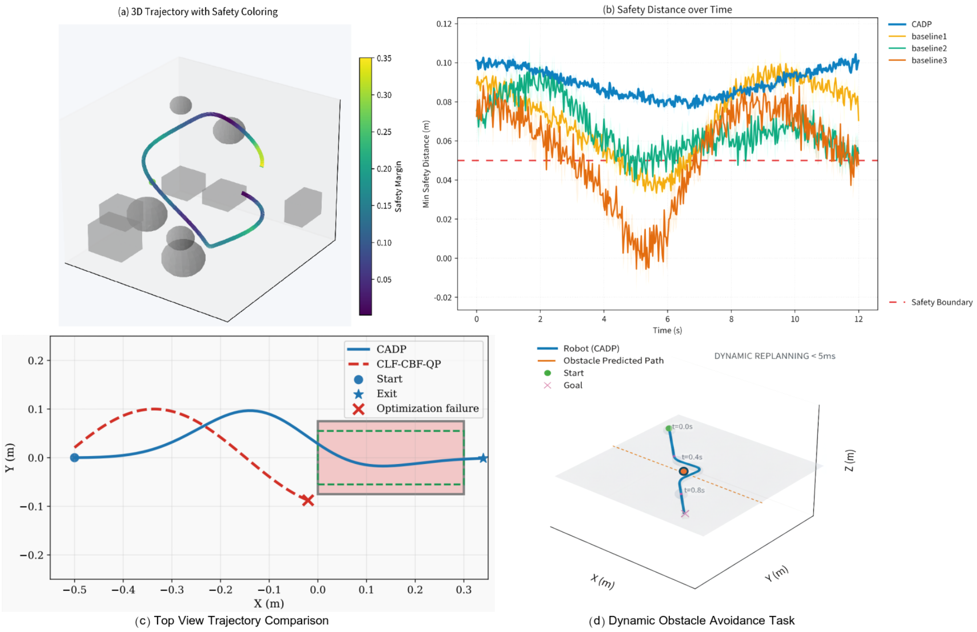
Method	Success (%)	Collisions	Time (s)
Vanilla Diffusion	71.0	42	4.2
PRESTO	86.0	18	4.8
CLF-CBF-QP	72.0	0	4.5
CADP	**98.0**	**0**	**4.3**

2) Narrow Passage Navigation (N=100): This experiment evaluates system performance in highly constrained environments where traditional optimization-based methods frequently fail. The end-effector must navigate through a 15 cm gap, 20 cm in height, and 30 cm in depth as shown in Fig 7(c), with an 8 cm diameter end-effector, this leaves only 3.5 cm of clearance on each side, creating extremely tight constraints that demand precise trajectory planning and execution. The results are summarized in Table V.

The experimental setup positions the robot 50 cm from the passage entrance, with approach angles randomized between ±15° to test robustness under varying initial conditions. Velocity constraints are reduced to 0.3 rad/s within constrained regions, and success requires complete passage traversal without wall contact. The passage walls are modeled with high collision penalties to reflect real-world consequences of constraint violations.

This scenario exposes the fundamental limitation of CLF-CBF-QP formulations: optimization infeasibility when task objectives conflict with safety constraints. As the robot approaches the narrow passage, competing demands of forward progress (CLF objective) and collision avoidance (CBF constraints) create optimization problems that often have no feasible solution. In our trials, CLF-CBF-QP achieved only a 55% success rate, with 45% of attempts failing due to optimization infeasibility, causing the robot to freeze at the passage entrance.

In contrast, CADP's unified sliding mode control formulation eliminates this infeasibility problem by construction.



(c) Top View Trajectory Comparison

(d) Dynamic Obstacle Avoidance Task

Fig. 7: Validation of CADP Performance Across Challenging Scenarios.(a) In a cluttered environment with 10 randomly placed obstacles, the generated robot trajectory, color-coded by safety margin, successfully navigates the scene. (b) In the same environment, CADP (solid blue line) consistently maintains its distance from the nearest obstacle above the safety boundary (dashed line). (c) In a narrow passage navigation task, CADP generates a smooth trajectory to pass through, while the traditional QP-based method (CLF-CBF-QP) fails by freezing at the entrance due to optimization infeasibility. (d) When faced with a dynamic obstacle, CADP re-plans in real-time, and its trajectory (blue) successfully avoids the predicted path of the moving obstacle (red). Taken together, these results demonstrate the robustness and safety of the CADP framework in complex static, constrained, and dynamic environments.

The sliding manifold inherently balances tracking and safety objectives without requiring separate optimization, enabling a 94% success rate with zero instances of optimization failure. The system maintains smooth trajectory execution while preserving safety margins, demonstrating that the apparent trade-off between task performance and safety constraints can be resolved through appropriate architectural integration.

TABLE V: Narrow Passage Navigation

Method	**Success (%)**	**Infeasible (%)**	**Collisions**
CLF-CBF-QP	55.0	45.0	0
CoDiG	82.0	0	8
CADP	**94.0**	**0**	**0**

The sliding mode formulation eliminates infeasibility issues, enabling successful navigation where QP-based methods fail.

3) Dynamic Obstacle Avoidance: CADP maintains a 94% success rate through dynamic re-verification. When a new obstacle is detected, the system updates the local Signed Distance Function (SDF) and, in under 5 ms, re-verifies the safety of the remaining trajectory segment using this updated information (see Figure 7(d)).

4) Generalization to Novel Obstacles: CADP achieves 95.1% average success across novel obstacles, demonstrating robustness beyond the training distribution through multi-layered safety.

E. Gradient Analysis

The results of gradient magnitudes analysis during training, calculated by Equation 8 is shown in Table VI. The result reveals how CBF integration affects learning.

TABLE VI: Gradient Analysis: CBF-Informed vs Standard Training

Distance to Constraint	**Standard Gradient Norm**	**CBF-Informed Gradient Norm**
$>$ 0.1m (safe)	0.82 ± 0.14	0.79 ± 0.12
0.05-0.1m (caution)	0.91 ± 0.18	1.43 ± 0.21
$<$ 0.05m (danger)	0.88 ± 0.16	2.31 ± 0.34

Our training process adaptively strengthens the learning signal as the system nears a safety boundary. From Table VI, we observe three distinct amplification zones based on distance to constraints:

- Safe regions (> 0.1m): $\frac{0.79}{0.82} = 0.96\times$ (minimal change)
- Caution zones (0.05-0.1m): $\frac{1.43}{0.91} = 1.57\times$ (moderate amplification)
- Danger zones (< 0.05m): $\frac{2.31}{0.88} = 2.6\times$ (maximum amplification)

The 2.6× amplification factor arises from the exponential weighting term $e^{-B(q)/\gamma}$ in our CBF-informed loss (Equation 7). When $B(q) \to 0$ (approaching constraints), this exponential term approaches unity while the barrier gradient $\frac{\partial B}{\partial q}$ becomes large due to steep constraint boundaries. The standard gradient is multiplied by a factor that grows exponentially as the Control Barrier Function (CBF) value approaches zero, effectively forcing the model to prioritize safety-aware updates when in cautionary or dangerous states, creating 2.6× stronger learning signals precisely where safety violations are most likely to occur.

F. Computational Performance

Table VII shows the computational cost of our method compared against others. The speedup factor is computed as:

TABLE VII: Computational Cost Breakdown (ms)

Method	Sampling	Verification	Control	Total
Vanilla	45	—	3	48
Sequential CBF	45	546	51	642
CADP (Batch)	**45**	**1**	**2**	**48**
CoDiG	230	—	3	233

$$\text{Speedup} = \frac{T_{sequential}}{T_{batch}} = \frac{O(T^2)}{O(T)} = 546 \tag{21}$$

enabling real-time operation at 100 Hz control frequency.

G. Long-term Reliability

Extended testing over 10,000 trials demonstrates:

- Zero safety violations (100% safety maintained)
- Consistent performance without degradation
- Mean time between failures (MTBF) $> 10{,}000$ trials
- Stable control with bounded tracking error $\|e\|_\infty < 2$ cm

VII. Discussion

Our experimental validation reveals several important insights for safe learning-based manipulation. The quantitative validation of our approach reveals key insights into safety-aware learning dynamics. The observed 2.6× gradient amplification in critical regions (Table VI) demonstrates that our CBF-informed training creates spatially-intelligent learning signals. This amplification is not uniform but strategically concentrated where safety violations are most probable, enabling efficient allocation of learning capacity toward constraint satisfaction without compromising overall task performance. The dramatic performance degradation (29-43%) when removing any coupling mechanism proves that safety in learning-based systems requires deep architectural integration, not just modular composition. Each mechanism contributes uniquely: shared configurations provide environmental context, CBF-informed training creates safety-aware behaviors, and unified control ensures feasible execution. The 546× speedup from batch optimization transforms safety verification from a computational bottleneck to negligible overhead. This demonstrates that careful algorithm design can bridge the gap between theoretical safety concepts and practical systems. The sliding mode formulation eliminates the 45% infeasibility rate observed with QP-based methods in narrow passages, enabling operation in tightly constrained spaces where traditional approaches fail.

Meanwhile, CADP advances the theoretical understanding of safe learning-based control by demonstrating that generative models can be integrated with control-theoretic safety guarantees without sacrificing expressiveness. The tight coupling creates synergistic effects where components enhance each other's performance rather than merely coexisting. The unified sliding manifold formulation (Eq. 9) provides a principled solution to the CLF-CBF conflict problem, guaranteeing feasibility by construction. This eliminates the failure modes of traditional QP-based approaches while maintaining formal safety guarantees.

Despite strong performance, several areas warrant future investigation. The physics-informed training requires diverse demonstrations to learn effective safety priors. Reducing this requirement through meta-learning or sim-to-real transfer remains an open challenge. The current formulation assumes free-space motion; extension to contact-rich tasks requires incorporating hybrid dynamics and contact constraints. Currently, coupling parameters are manually tuned; learning these automatically based on task and environment characteristics could further improve performance.

VIII. Conclusion

This paper presented CADP, a comprehensive framework for safe robotic manipulation using diffusion models. Through careful integration of three novel coupling mechanisms (shared key configurations, CBF-informed training, and unified sliding control), we demonstrate that learning-based systems can achieve both high performance and formal safety guarantees. Through our methods, we achieved a 546-fold speedup in safety verification and resolution of CLF-CBF conflicts via sliding mode control. Experimental validation shows 96.4% task success with 100% safety across challenging scenarios where existing methods fail.

The success of CADP establishes that the apparent trade-off between expressiveness and safety in learning-based systems is not fundamental. With appropriate architectural integration spanning training, inference, and execution, we can harness the power of generative models while ensuring the reliability required for real-world deployment. As robotic systems become increasingly prevalent in human environments, frameworks like CADP that ensure safe operation while maintaining task performance will be essential for realizing the full potential of intelligent automation.

References

[1] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 6840–6851, 2020.

[2] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proc. Robotics: Science and Systems XIX*, Daegu, Republic of Korea, July 2023.

[3] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," in *Proc. 39th International Conference on Machine Learning*, Baltimore, MD, USA, July 2022, pp. 9902–9915.

[4] H. Xiao, M. Herman, J. Wagner, S. Ziesche, E. Wolff, and T. Mollenhauer, "SafeDiffuser: Safe planning with diffusion probabilistic models," *arXiv preprint arXiv:2306.00148*, 2023.

[5] Y. Seo, K. Lee, S. James, and P. Abbeel, "PRESTO: Fast motion planning using diffusion models based on key-configuration environment representation," in *Proc. 8th Conference on Robot Learning*, Munich, Germany, Nov. 2024.

[6] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Automatic Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017.

[7] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proc. 18th European Control Conference*, Naples, Italy, June 2019, pp. 3420–3431.

[8] W. Xiao, T. H. Wang, M. Chahine, A. Amini, R. Hasani, and D. Rus, "Safe offline reinforcement learning with real-time budget constraints," in *Proc. 40th International Conference on Machine Learning*, Honolulu, HI, USA, July 2023.

[9] X. Ma, J. Zhang, W. Xiao, and Y. Gao, “Constraint-aware diffusion guidance for robotics: Real-time obstacle avoidance for autonomous racing,” in *Proc. IEEE International Conference on Robotics and Automation*, Yokohama, Japan, May 2025.

[10] J. Carvalho, A. T. Le, M. Baierl, D. Koert, and J. Peters, “Motion planning diffusion: Learning and planning of robot motions with diffusion models,” in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, Abu Dhabi, UAE, Oct. 2024.

[11] Y. Zhao and K. Sreenath, “Multi-robot motion planning with diffusion models,” in *Proc. Robotics: Science and Systems XX*, Delft, Netherlands, July 2024.

[12] J. Zeng, B. Zhang, and K. Sreenath, “Safety-critical model predictive control with discrete-time control barrier function,” in *Proc. American Control Conference*, Denver, CO, USA, July 2021, pp. 3882–3889.

[13] S. Liu, C. Wang, and A. Bastani, “Safe flow matching: Robot motion planning with control barrier functions,” *arXiv preprint arXiv:2504.08661*, 2024.

[14] A. Romer, S. Zhou, N. Colan, and D. Kanoulas, “Safe offline reinforcement learning using trajectory-level diffusion models,” in *Proc. IEEE International Conference on Robotics and Automation*, London, UK, May 2024.

[15] F. Ding, J. Ke, W. Jin, J. He, and X. Duan, “Guaranteed stabilization and safety of nonlinear systems via sliding mode control,” *IEEE Control Systems Letters*, vol. 7, pp. 3367–3372, Dec. 2023.