Inference-Time Preference-Aligned Diffusion Planning for Safe Offline Reinforcement Learning

Weichen Li * (✉), Waleed Mustafa *, Puyu Wang, Marius Kloft, and Sophie Fellenz

University of Kaiserslautern-Landau, Germany
weichen@cs.uni-kl.de

Abstract. Safe Reinforcement Learning (Safe RL) aims to train agents that maximize expected rewards while adhering to safety constraints, often modeled as costs. However, in real-world applications, the definitions of desirable and undesirable behavior are inherently context-dependent and subject to change over time. To address this challenge, we propose a flexible and modular offline Safe RL framework that decouples reward and cost optimization using separate diffusion policy models trained via Direct Preference Optimization (DPO). Unlike existing approaches that combine multiple objectives into a single model or require retraining when objectives change, our method independently learns reward and cost components. At inference time, these components are composed with adjustable weightings, enabling users to adapt the agent's behavior dynamically based on contextual preferences—without the need for retraining. This compositional strategy supports alignment with evolving reward specifications while significantly reducing computational overhead. We empirically validate the effectiveness of our method on standard offline Safe RL benchmark, demonstrating robust performance.

Keywords: Preference Learning · Decision Making · Safe Reinforcement Learning

1 Introduction

In safety-critical domains such as autonomous driving [21], and robotics [13, 12], reinforcement learning (RL) agents must optimize not only for task success but also for safety. This challenge is the focus of Safe RL, which augments standard RL by introducing cost functions that penalize unsafe behaviors such as collisions, excessive forces, or violations of operational constraints.

Most Safe RL approaches formulate this as a constrained optimization problem, where the agent seeks to maximize the expected reward while ensuring the expected cost remains below a fixed threshold. While effective in controlled settings, this rigid formulation often lacks the flexibility required for real-world deployment. In practice, the trade-off between performance and safety is rarely constant—it varies across users, tasks, and environments.

* Equal contribution.

Consider autonomous driving: regulations and preferences can change over time. For example, a law might increase the minimum distance allowed when passing a pedestrian from 1 meter to 1.5 meters. Or a new speed limit may be introduced on familiar roads. The goal of reaching your destination efficiently remains unchanged, but the constraints on how to do so evolve. Human drivers naturally adapt to such changes without retraining. Similarly, RL agents should be capable of adapting their behavior to new or updated safety constraints, without requiring retraining. In this paper, we propose a framework in which, at inference time, users can specify a set of weights over objectives to express their current priorities. The policy then combines individual objective-aligned models accordingly. This approach eliminates the need to learn joint preferences during training and instead supports flexible, real-time preference composition. Furthermore, new objectives can be integrated seamlessly by training individual models independently, without retraining the full system.

While inference-time preference composition offers flexibility, it also poses a key challenge for training: how do we ensure that individual models learn behaviors that are both distinct and composable, especially without relying on explicit reward or cost signals? To this end, we introduce a diffusion-based Safe RL framework trained via Direct Preference Optimization (DPO) [18]. Rather than requiring explicit scalar rewards, DPO leverages pairwise preferences—such as "trajectory A is safer than trajectory B"—to guide learning. We use diffusion models to learn rich trajectory distributions, making it possible to train high-capacity policies from offline data and enable nuanced preference modeling.

Moreover, this unified framework naturally supports varying user preferences across contexts. Different users or applications may weigh performance and safety differently, and our system allows agents to align their behavior accordingly—without retraining, and without access to dense reward/cost labels.

Our main contributions are as follows:

- We propose an inference-time alignment method for Safe RL that allows dynamic, user-defined reward and cost weighting. Moreover, our approach enables new objectives (e.g., safety) to be incorporated without retraining the entire model—only the component corresponding to the new objective requires training.
- To enable inference-time alignment, we develop a diffusion-based Safe RL framework using Direct Preference Optimization (DPO), which learns from pairwise preferences without requiring explicit reward or cost signals. Experiments on standard offline Safe RL demonstrate the effectiveness of the proposed method.

2 Related Work

2.1 Diffusion Model for Reinforcement Learning

Diffusion models have emerged as a powerful tool for sequential decision-making, traditionally the domain of RL. Two primary paradigms have surfaced: diffusion

policy and diffusion planning [9, 25]. In the former, the diffusion model is used as a policy to generate individual actions, updated via standard RL algorithms [20]. In the latter, the model plans full state trajectories in a single forward pass, bypassing environment interaction during execution [1, 16]. In this work, we adopt the diffusion planning paradigm to evaluate its effectiveness in a fully offline RL setting, emphasizing its benefits in trajectory generation, safety, and performance without requiring environmental interaction during training.

2.2 Safe Reinforcement Learning

Lagrangian optimization has been widely used to train reinforcement learning (RL) agents that prioritize safety, in both online [23, 6] and offline [8, 10] settings. However, Lagrangian-based methods typically require access to both reward and cost signals, and identifying the optimal Lagrange multiplier can be challenging in practice. More recently, diffusion models have been introduced into the offline RL paradigm, where learning is performed solely from fixed datasets. Building on the Diffuser framework, several approaches reformulate offline RL as a conditional sequence generation problem using denoising diffusion probabilistic models (DDPMs). Zheng et al. enforce hard constraints by defining a feasibility set guided by value functions to ensure zero policy violations [24]. Lin et al. introduce soft budget constraints through a dual critic architecture combined with a Lagrangian formulation [11].

In contrast, OASIS addresses the challenge of suboptimal or unsafe offline datasets by re-sampling high-return, safe trajectories using a conditional diffusion model, followed by standard offline RL training on the augmented dataset [22]. In our work, we adopt the core idea of OASIS but extend it by directly optimizing the diffusion model using DPO, thereby removing the need for conditioning mechanisms.

3 Background

The notation used throughout the remainder of this paper is defined as follows. A trajectory $\tau = (S, A)$ consists of a sequence of states $S = (s_1, s_2, \ldots, s_T)$ and actions $A = (a_1, a_2, \ldots, a_T)$. Each trajectory is associated with a final reward and cost score. For preference learning, we construct a trajectory pair dataset, represented as $(\tau_i, \tau_j) \in D_{\text{pref}}$, where each pair consists of two trajectories of the same length. Safe RL is typically formulated as a constrained optimization problem.

3.1 Offline Constrained RL

The Constrained Markov Decision Process (CMDP) [2] is defined as: $M := (\mathcal{S}, \mathcal{A}, \mathcal{T}, \gamma, r, c)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{T}$ is the transition probability function, $\gamma \in (0, 1]$ is the discount factor, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and $c : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ denote the reward and cost functions respectively. In offline RL,

the policy π is trained from the offline dataset D_{pref}, relying on historical data rather than active interaction with the environment to learn effective behavior.

Most existing work in Safe Reinforcement Learning formulates the problem as finding a policy π that maximizes the expected discounted sum of rewards, augmented by an entropy regularization term while satisfying multiple cost constraints and does not exceed a predefined threshold u_i. Specifically, the problem is formulated as

$$\begin{aligned} \max_{\pi} \quad & \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=1}^{T}\gamma^t\Big(r(s_t,a_t)+\alpha\mathcal{H}(\pi(\cdot\mid s_t))\Big)\right] \\ \text{s.t.} \quad & \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=1}^{T}\gamma^t c_i(s_t,a_t)\right]\le u_i, \quad i=1,\dots,k, \\ & \sum_{a\in\mathcal{A}}\pi(a\mid s)=1 \quad \forall s\in\mathcal{S}, \end{aligned} \tag{1}$$

where,

- $\pi:\mathcal{S}\to\Delta(\mathcal{A})$ is the policy, mapping from a state to a probability distribution over $a\in\mathcal{A}$.
- u_i is the predefined threshold for constraints.
- $\alpha>0$ is the entropy regularization coefficient.

The above formulation balances two competing objectives: maximizing the long-term rewards while ensuring the agent's behavior is constrained within acceptable cost limits. However, a notable limitation is that the cost thresholds u must be specified in advance. If these thresholds change, the policy typically requires retraining. Furthermore, in many tasks, well-defined cost boundaries are unavailable or ambiguous, and the model's performance can be highly sensitive to the choice of threshold values.

4 Inference-time Alignment

In this section, we propose an inference-time alignment method for Safe RL. Denote $w=[w_r,w_c]\in\mathbb{R}^2$ the weight vector, where w_r and w_c represent the relative importance of reward maximization and cost minimization (i.e., safety), respectively.

Motivated by recent advances in diffusion-based offline RL, we propose modeling the distribution over entire trajectories $\tau=\{(s_t,a_t)\}_{t=1}^{T}$ directly. To decouple performance and safety, we first train two separate trajectory distributions: one that optimizes for reward and another for cost. At inference time, we combine these distributions into a scalarized form using a weighted geometric mean, where user-specified weights w_r and w_c control the trade-off between performance and safety:

$$p^\star(\tau)\propto\left(p_r^\star(\tau)\right)^{w_r}\cdot\left(p_c^\star(\tau)\right)^{w_c}, \tag{2}$$

where the optimal trajectory distributions for reward and cost are defined as:

$$p_r^\star = \arg\max_p \ \mathbb{E}_{\tau\sim P}\left[\sum_{t=1}^{T}\gamma^t r(s_t,a_t)\right] - \beta\,\mathrm{KL}(p\|p_{\mathrm{ref}}),$$

$$p_c^\star = \arg\min_p \ \mathbb{E}_{\tau\sim p}\left[\sum_{t=1}^{T}\gamma^t c(s_t,a_t)\right] - \beta\,\mathrm{KL}(p\|p_{\mathrm{ref}}),$$

where p_{ref} is a reference trajectory distribution.

This formulation enables flexible trade-offs between performance and safety. When $w_c = 0$, the objective reduces to pure reward maximization; when $w_r = 0$, it corresponds to strict safety optimization.

Based on this insight, we propose an approach where the reward-based distribution $p_r^\star(\tau)$ and the cost-based distribution $p_c^\star(\tau)$ are learned independently. At test time, given a user-defined preference vector w, we sample from the scalarized distribution defined above.

A key benefit of this formulation is its support for **incremental objective discovery**. If cost information is unavailable during initial training, we can simply set $w_c = 0$. Later, when cost becomes observable (e.g., during deployment or continual learning), it is sufficient to train only the cost-based distribution $p_c^\star(\tau)$ independently. The user can then update the preference vector by assigning a nonzero weight to w_c, incorporating the new objective without retraining the reward-based component $p_r^\star(\tau)$.

5 Diffusion-based Safe RL Framework

Diffusion models have recently demonstrated strong performance in generative modeling and sequential decision-making tasks, owing to their expressive capacity and flexibility [1, 16, 22]. To operationalize our alignment framework, we employ a diffusion model as the backbone for trajectory generation. The overall procedure is summarized in Algorithm 1, and consists of three primary stages:

1. **Pretraining:** Learning a diffusion-based planner to model the distribution over environment trajectories in a task-agnostic manner.
2. **Preference-guided Fine-tuning:** Fine-tune separate diffusion models for each objective using pairwise trajectory preferences. These preferences are derived from comparisons of total value scores, favoring trajectories with higher rewards or lower costs, depending on the specific objective.
3. **Inference time Alignment:** At test time, based on Equation (2), user-defined preference weights over objectives are used to interpolate between fine-tuned planners, enabling dynamic and flexible control over the trade-off between reward and cost.

5.1 Pre-trained Diffusion Model as Planner

We first train a diffusion model using the entire offline dataset to learn a general-purpose planner, disregarding any safety considerations. This model, which we refer to as the **reference model**, captures the trajectory distribution of behavior policies without any bias towards safe or unsafe actions. Following prior work on denoising diffusion probabilistic models [7, 1, 22], the model learns to denoise corrupted sequences $\tau = [s_0, \ldots, s_T]$ by predicting noise vectors using the standard loss:

$$\mathcal{L}(\theta) = \mathbb{E}_{x_0,t,\epsilon}\left[\left\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\right\|_2^2\right],$$

where x_0 denotes the original (uncorrupted) trajectory representation, t is the diffusion timestep, and $\epsilon \sim \mathcal{N}(0, I)$ is the noise added during the forward diffusion process. The model ϵ_θ is trained to predict the noise component at each timestep.

Inverse Dynamic Model. To recover actions from predicted states, we train an auxiliary inverse dynamics model $f_\phi(s_{t+m}, s_t)$ that infers actions from state pairs. Here, m denotes the step stride between two states. This decouples trajectory planning in state space from control in action space, improving modularity and sample efficiency.

5.2 DPO-based Diffusion Model Fine-tuning

In the second stage of our framework, we fine-tune two separate diffusion models, one for maximizing the reward and one for maximizing the safety using pairwise trajectory preferences. These preferences are derived from comparisons of scalarized trajectory returns—prioritizing higher cumulative rewards or lower cumulative costs, respectively. The goal is to align each diffusion model with its corresponding objective by increasing the likelihood of preferred trajectories while maintaining proximity to a pre-trained reference distribution.

Direct Preference Optimization (DPO). In pairwise preference learning, the goal is to model user preferences of the form $\tau^+ \succ \tau^-$, indicating that trajectory τ^+ is preferred over τ^-. A foundational assumption in this setting is the *Bradley–Terry model* [3], which defines the probability of preferring τ^+ over τ^- using latent utility (or reward) scores:

$$p(\tau^+ \succ \tau^-) = \frac{\exp\left(r^*(\tau^+)\right)}{\exp\left(r^*(\tau^+)\right) + \exp\left(r^*(\tau^-)\right)},$$

where $r^*(\cdot)$ is a latent reward function that assigns scalar utility to each trajectory. Rather than explicitly modeling r^*, DPO directly optimizes a policy to increase the log-probability of preferred trajectories, using this probabilistic model as an objective [17].

Algorithm 1 Preference-Guided Diffusion Policy

Input: Offline dataset $\mathcal{D}$, Pairwise dataset $\mathcal{D}_{pref}$, initialized diffusion policy ϵ_θ
Output: Fine-tuned policies $\{\epsilon_{\theta_i}\}$
Pretraining: Diffusion
for each clean sample $x_0 \in \mathcal{D}$, timestep $t \sim \mathcal{U}[1, T]$, noise $\epsilon \sim \mathcal{N}(0, I)$ **do**
    $x_t \leftarrow \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1 - \bar{\alpha}_t}\, \epsilon$ ▷ Noisy version of x_0 via forward diffusion
    $\mathcal{L}_{\text{diff}} \leftarrow \|\epsilon - \epsilon_\theta(x_t, t)\|^2$
end for
Fine-Tuning: Preference-Based Optimization
for each preference pair $(x_0^{(w)}, x_0^{(l)}) \sim \mathcal{D}_p$ **do**
    Compute log-likelihoods under current model and reference:

$$\mathcal{L}_{\text{DPO-Diffusion}}(\theta) = -\mathbb{E}_{(x_0^{(+)}, x_0^{(-)})} \left[\log \sigma \left(\beta \Big(\log \frac{p_\theta(x_0^{(+)})}{p_{\text{ref}}(x_0^{(+)})} - \log \frac{p_\theta(x_0^{(-)})}{p_{\text{ref}}(x_0^{(-)})} \Big) \right) \right]$$

end for
Inference: Preference-Controlled Sampling
Sample $x_T \sim \mathcal{N}(0, I)$ ▷ Initial latent
for $t = T, T-1, \ldots, 1$ **do**
    Define ensemble denoiser with preference weights $\{w_i\}$:

$$\epsilon(x_t, t) = \sum_i w_i \cdot \epsilon_{\theta_i}(x_t, t)$$

    Predict clean sample: $\hat{x}_0 = \text{Predict}(x_t, t, \epsilon(x_t, t))$
    Sample x_{t-1} from $p(x_{t-1} \mid x_t, \hat{x}_0)$ ▷ Reverse diffusion step
end for
Return generated sample x_0

DPO for Diffusion Model Optimization. To adapt DPO to diffusion models, we treat the trajectory log-likelihood under the model as a proxy for trajectory utility. Given a pair of trajectories (τ^+, τ^-) such that $\tau^+ \succ \tau^-$, we use the DPO objective tailored to generative diffusion models [19]:

$$L_{\text{DPO-Diffusion}}(\theta) = -\mathbb{E}_{(x_0^+, x_0^-)} \log \sigma \left(\beta \log \frac{p_\theta(x^+)}{p_{\text{ref}}(x^+)} - \log \frac{p_\theta(x^-)}{p_{\text{ref}}(x^-)} \right), \tag{3}$$

where p_θ and p_{ref} denote trajectory likelihoods under the current and reference models, respectively. This objective encourages the model to increase the probability of preferred trajectories while staying close to the pretrained distribution.

5.3 Inference Time Multi-Objective Alignment

At test time, user-defined preferences $w = [w_r, w_c]$ are used to compose a denoising function as a convex combination of single-objective models:

$$\epsilon(x, t) = \sum_{i=1}^{m} w_i \cdot \epsilon_{\theta_i}(x, t), \tag{4}$$

where each ϵ_{θ_i} is trained independently for objective i, such as a reward or cost model. This interpolation mechanism enables preference-controlled generation without requiring the model to observe or condition on w during training.

If a reward or a cost component is unavailable during training, the corresponding distribution can be learned independently at a later stage. It can then be incorporated at test time by assigning a nonzero preference weight $w_j > 0$, without requiring retraining of the previously learned components.

6 Experiments

In this section, we evaluate our method on the Bullet-Safety-Gym benchmark [5], a suite of environments designed to test safety-aware decision-making in reinforcement learning.

6.1 Environments: Bullet Safety Tasks

We evaluate six tasks from the Bullet-Safety-Gym benchmark, organized into two categories: **Circle Tasks** — *BallCircle*, *CarCircle*, and *DroneCircle*; and **Run Tasks** — *BallRun*, *CarRun*, and *DroneRun*.

These tasks challenge agents to complete navigation objectives while minimizing safety violations, such as leaving a safe region or colliding with obstacles. This makes them particularly well-suited for evaluating preference-conditioned planning, where reward-safety trade-offs must be respected at inference time.

6.2 Offline Dataset Preparation

For offline RL, we follow the DSRL dataset proposed by Liu et al. [15]. This offline dataset is used to pretrain a general-purpose diffusion model.

Pairwise Data Generation. To generate the pairwise data required for DPO fine-tuning, we sample trajectory pairs from the offline dataset. For each pair, we randomly select two trajectories and compute the cumulative rewards and costs over sub-trajectories. These aggregated values are then used to establish pairwise preferences, which serve as supervision for aligning the diffusion model with specific objectives.

6.3 Evaluation Metrics

We evaluate agent performance using two primary metrics: reward and cost. A higher reward indicates better performance, while a lower cost indicates greater safety. The ground truth reward and cost during evaluation are received from the environment; their definitions are as follows. When the normalized cost is smaller than 1, the agent is defined as a safe agent [22, 24].

Circle Tasks. The reward function for circle tasks encourages agents to move along the circular boundary and is defined as:

$$r(s, a, s') = \frac{-yv_x + xv_y}{1 + \left|\sqrt{x^2 + y^2} - radius\right|} + r_{\text{robot}}(s).$$

The cost function penalizes agents for exceeding the circular boundary and is defined as:

$$c(s_t) = \mathbf{1}(|x| > x_{\text{lim}}).$$

Run Tasks. The reward function encourages agents to move quickly in a specific direction and is defined as:

$$r(s, a, s') = \|x_{t-1} - g\|_2 - \|x_t - g\|_2 + r_{\text{robot}}(s_t).$$

The cost function penalizes agents for crossing the vertical boundary or exceeding a velocity threshold:

$$c(s, a, s') = \max\left(1, \mathbf{1}(|y| > y_{\text{lim}}) + \mathbf{1}(\|v_t\|_2 > v_{\text{lim}})\right).$$

Notably, no reward or cost is required during training since we use DPO.

6.4 Preference-Based Control via Weight Alignment

We assess whether our model supports post-training preference alignment—the ability to adjust reward-cost trade-offs at inference time without retraining.

As shown in Table 1, our approach enables dynamic adjustment of the reward-cost balance by assigning different weights to objectives during planning. The user provides a preference vector $(w_{\text{reward}}, w_{\text{cost}})$ where $w_{\text{reward}} + w_{\text{cost}} = 1$. These weights modulate the classifier-free guidance at generation time using Equation 4.

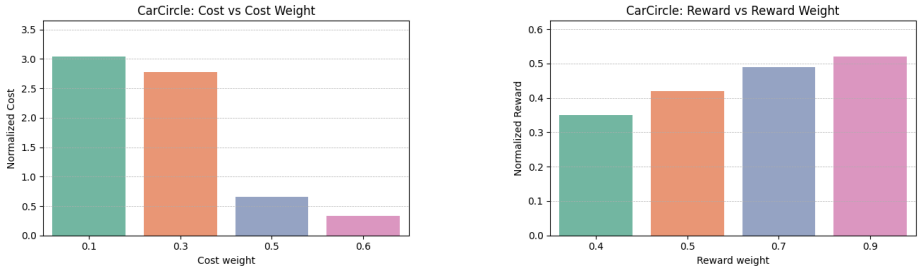
When w_{reward} is high (e.g., 0.9), the agent prioritizes achieving the task goal, often at the expense of accumulating more cost. Conversely, increasing the cost weight w_{cost} encourages the agent to behave more conservatively, minimizing safety violations even if it means achieving lower task rewards. Importantly, our model demonstrates a smooth interpolation between these two extremes, effectively balancing reward-seeking and cost-avoiding behaviors through conditional guidance during inference.

6.5 Comparison with Prior Work: DPO-Based Diffusion Policy

To evaluate the effectiveness of our approach, we compare it with a diverse set of strong offline Safe RL baselines, including behavior cloning (BC), cost-constrained RL (CPQ, COptiDICE), and preference-conditioned diffusion or generative models (CDT, CVAE-BCQL, OASIS). The results presented in Table 2 are based on the experimental findings reported by [22].

Task	w =(w_r, w_c)	Reward (r)	Cost (c)	Preference Bias
CarCircle	(0.9, 0.1)	0.52	3.04	Reward-focused
CarCircle	(0.7, 0.3)	0.49	2.78	Slight reward bias
CarCircle	(0.5, 0.5)	0.42	0.65	Balanced
CarCircle	(0.4, 0.6)	0.35	0.33	Cost-focused
CarRun	(0.9, 0.1)	0.92	1.69	Reward-focused
CarRun	(0.7, 0.3)	0.86	0.35	Slight reward bias
CarRun	(0.5, 0.5)	0.85	0.00	Balanced

Table 1: Effect of inference-time reward–cost weighting on agent performance across tasks. Higher reward weights result in increased normalized reward at the expense of higher cost, demonstrating a trade-off.



(a) Normalized cost values for different cost weights across tasks.

(b) Normalized reward values for different reward weights across tasks.

Fig. 1: Comparison of agent performance metrics under varying weighting schemes for cost and reward. Each subfigure shows how the respective metric changes with different weight settings. Increasing the weight of cost encourages the agent to pay more attention to minimizing cost, thus lowering cost values. Conversely, increasing the weight of reward results in higher reward values.

Our DPO-based diffusion model consistently demonstrates strong performance across both objectives. Remarkably, it achieves safe behavior—defined as maintaining a normalized cost below 1 [14]—across all tasks. Unlike prior methods (except BC), which depend on reward and cost signals during training, our approach operates without requiring access to immediate rewards or costs.

During inference, we evaluate various combinations of reward-cost weights to examine how well different trade-offs balance performance and safety—without requiring retraining. Example weight configurations include $(0.1, 0.9), (0.5, 0.5)$, and $(0.9, 0.1)$. If the observed cost exceeds a predefined threshold (e.g., 1.0), we increase the weight assigned to cost to promote safer behavior. Conversely, if the cost remains below the threshold, we compare the reward with a baseline (e.g., OASIS) and attempt to increase the reward weight while maintaining safety. For each combination, we run experiments using three different random seeds. For each seed, we compute the average result over 20 episodes. The final

Algorithm	Stats	BallRun	CarRun	DroneRun	BallCircle	CarCircle	DroneCircle
BC	reward ↑	0.55 ± 0.23	0.94 ± 0.02	0.62 ± 0.11	0.73 ± 0.05	0.59 ± 0.11	0.82 ± 0.01
	cost ↓	2.04 ± 1.32	1.50 ± 1.11	3.48 ± 0.68	2.53 ± 0.15	3.39 ± 0.85	3.29 ± 0.18
CPQ	reward ↑	0.25 ± 0.11	0.63 ± 0.51	0.13 ± 0.30	**0.39 ± 0.34**	**0.64 ± 0.02**	0.01 ± 0.02
	cost ↓	1.34 ± 1.32	1.43 ± 1.82	2.29 ± 1.98	**0.73 ± 0.66**	**0.12 ± 0.19**	3.16 ± 3.85
COptiDICE	reward ↑	0.63 ± 0.04	**0.90 ± 0.03**	0.71 ± 0.01	0.73 ± 0.02	0.52 ± 0.01	**0.35 ± 0.02**
	cost ↓	3.13 ± 0.17	**0.28 ± 0.24**	3.87 ± 0.08	2.83 ± 0.23	3.56 ± 0.16	**0.12 ± 0.10**
BEAR-Lag	reward ↑	0.65 ± 0.08	0.55 ± 0.62	0.10 ± 0.33	0.89 ± 0.02	0.80 ± 0.08	0.89 ± 0.04
	cost ↓	4.38 ± 0.28	8.44 ± 0.62	3.72 ± 3.22	2.84 ± 0.28	2.89 ± 0.84	4.03 ± 0.51
BCQ-Lag	reward ↑	0.51 ± 0.19	0.96 ± 0.06	0.76 ± 0.07	0.76 ± 0.04	0.79 ± 0.02	0.88 ± 0.04
	cost ↓	1.96 ± 0.88	2.31 ± 3.22	5.19 ± 1.08	2.62 ± 0.29	3.25 ± 0.28	3.90 ± 0.55
CDT	reward ↑	0.35 ± 0.01	0.96 ± 0.01	0.84 ± 0.12	0.73 ± 0.01	0.71 ± 0.01	0.17 ± 0.08
	cost ↓	1.56 ± 1.10	**0.67 ± 0.03**	7.56 ± 0.33	1.36 ± 0.03	2.39 ± 0.15	1.08 ± 0.62
FISOR	reward ↑	**0.17 ± 0.03**	0.85 ± 0.02	0.44 ± 0.14	**0.28 ± 0.03**	**0.24 ± 0.05**	**0.49 ± 0.05**
	cost ↓	**0.04 ± 0.06**	0.15 ± 0.20	2.52 ± 0.61	**0.00 ± 0.00**	**0.15 ± 0.27**	**0.02 ± 0.03**
CVAE-BCQL	reward ↑	0.25 ± 0.02	**0.88 ± 0.05**	0.21 ± 0.52	**0.49 ± 0.03**	**0.60 ± 0.05**	0.01 ± 0.02
	cost ↓	1.40 ± 0.35	**0.00 ± 0.00**	2.80 ± 0.63	1.39 ± 0.27	1.77 ± 0.47	3.31 ± 1.66
OASIS	reward ↑	**0.28 ± 0.01**	**0.85 ± 0.04**	**0.13 ± 0.08**	0.70 ± 0.01	0.76 ± 0.03	0.60 ± 0.01
	cost ↓	**0.79 ± 0.37**	**0.02 ± 0.03**	**0.79 ± 0.54**	**0.45 ± 0.14**	**0.89 ± 0.59**	**0.25 ± 0.10**
Our Results	reward ↑	0.30±0.03	**0.90±0.003**	0.27±0.01	0.70±0.00	**0.42 ±0.0**	**0.16 ±0.0**
	cost ↓	**0.92±0.22**	**0.01 ± 0.01**	**0.00 ± 0.0**	**0.94 ± 0.03**	**0.75 ±0.1**	**0.76 ±0.01**

Table 2: Evaluation results of the normalized reward and cost. The cost threshold is 1. ↑: the higher the reward, the better. ↓: the lower the cost (up to threshold 1), the better. **Bold**: Safe agents whose normalized cost is smaller than 1. Blue: Safe agent with the highest reward.

performance metrics are then averaged across the three seeds. We report the results corresponding to the best-performing reward-cost trade-off. For example, in the OfflineDroneRun-v0 task with reward and cost weights of 0.1 and 0.9 respectively, our method records a cost of precisely 0.00 while maintaining a competitive reward of 0.27, which is higher than many safety-focused baselines. In the OfflineCarRun-v0 task, our model achieves a near-optimal reward of 0.90 while maintaining an extremely low cost of 0.01.

7 Limitations and Future Work

While our method demonstrates promising results in the domain of safe offline reinforcement learning, several directions for future research and known limitations warrant further investigation. First, although we have shown strong performance on Bullet-Safety-Gym tasks, our evaluation is currently limited to safety-oriented environments. A natural extension of this work is to apply our approach to broader multi-objective reinforcement learning (MORL) benchmarks [4], such as those based on the MuJoCo simulator. These environments typically feature more diverse and fine-grained trade-offs among objectives (e.g., speed vs. energy consumption vs. stability), which would allow us to further test the flexibility and generalizability of our diffusion-based preference planning framework.

Second, model behavior warrants deeper interpretability analysis. While our method consistently ensures safe behavior across all tasks, it does not always achieve optimal reward outcomes. For instance, in the 'DroneCircle' task, the

model satisfies safety constraints but fails to maximize the reward. A more detailed analysis of such cases is necessary to understand potential limitations in preference alignment and reward optimization.

References

1. Ajay, A., Du, Y., Gupta, A., Tenenbaum, J., Jaakkola, T., Agrawal, P.: Is conditional generative modeling all you need for decision-making? arXiv preprint arXiv:2211.15657 (2022)
2. Altman, E.: Constrained Markov decision processes. Routledge (2021)
3. Bradley, R.A., Terry, M.E.: Rank analysis of incomplete block designs: I. the method of paired comparisons. Biometrika **39**(3/4), 324–345 (1952)
4. Felten, F., Alegre, L.N., Nowe, A., Bazzan, A., Talbi, E.G., Danoy, G., C da Silva, B.: A toolkit for reliable benchmarking and research in multi-objective reinforcement learning. Advances in Neural Information Processing Systems **36**, 23671–23700 (2023)
5. Gronauer, S.: Bullet-safety-gym: A framework for constrained reinforcement learning. Tech. rep., mediaTUM (2022). https://doi.org/10.14459/2022md1639974
6. Gu, S., Yang, L., Du, Y., Chen, G., Walter, F., Wang, J., Knoll, A.: A review of safe reinforcement learning: methods, theories and applications. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
7. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. Advances in neural information processing systems **33**, 6840–6851 (2020)
8. Hong, K., Li, Y., Tewari, A.: A primal-dual-critic algorithm for offline constrained reinforcement learning. In: International Conference on Artificial Intelligence and Statistics. pp. 280–288. PMLR (2024)
9. Janner, M., Du, Y., Tenenbaum, J.B., Levine, S.: Planning with diffusion for flexible behavior synthesis. arXiv preprint arXiv:2205.09991 (2022)
10. Le, H., Voloshin, C., Yue, Y.: Batch policy learning under constraints. In: International Conference on Machine Learning. pp. 3703–3712. PMLR (2019)
11. Lin, Q., Tang, B., Wu, Z., Yu, C., Mao, S., Xie, Q., Wang, X., Wang, D.: Safe offline reinforcement learning with real-time budget constraints. In: International Conference on Machine Learning. pp. 21127–21152. PMLR (2023)
12. Liu, P., Tateo, D., Ammar, H.B., Peters, J.: Robot reinforcement learning on the constraint manifold. In: Conference on Robot Learning. pp. 1357–1366. PMLR (2022)
13. Liu, P., Zhang, K., Tateo, D., Jauhri, S., Hu, Z., Peters, J., Chalvatzaki, G.: Safe reinforcement learning of dynamic high-dimensional robotic tasks: navigation, manipulation, interaction. In: 2023 IEEE International Conference on Robotics and Automation (ICRA). pp. 9449–9456. IEEE (2023)
14. Liu, Z., Guo, Z., Lin, H., Yao, Y., Zhu, J., Cen, Z., Hu, H., Yu, W., Zhang, T., Tan, J., Zhao, D.: Datasets and benchmarks for offline safe reinforcement learning. Journal of Data-centric Machine Learning Research (2024)
15. Liu, Z., Guo, Z., Lin, H., Yao, Y., Zhu, J., Cen, Z., Hu, H., Yu, W., Zhang, T., Tan, J., et al.: Datasets and benchmarks for offline safe reinforcement learning. arXiv preprint arXiv:2306.09303 (2023)
16. Lu, H., Han, D., Shen, Y., Li, D.: What makes a good diffusion planner for decision making? In: The Thirteenth International Conference on Learning Representations (2025)

17. Rafailov, R., Sharma, A., Mitchell, E., Manning, C.D., Ermon, S., Finn, C.: Direct preference optimization: Your language model is secretly a reward model. Advances in Neural Information Processing Systems **36**, 53728–53741 (2023)
18. Rafailov, R., Sharma, A., Mitchell, E., Manning, C.D., Ermon, S., Finn, C.: Direct preference optimization: Your language model is secretly a reward model. Advances in Neural Information Processing Systems **36** (2024)
19. Wallace, B., Dang, M., Rafailov, R., Zhou, L., Lou, A., Purushwalkam, S., Ermon, S., Xiong, C., Joty, S., Naik, N.: Diffusion model alignment using direct preference optimization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8228–8238 (2024)
20. Wang, Z., Hunt, J.J., Zhou, M.: Diffusion policies as an expressive policy class for offline reinforcement learning. arXiv preprint arXiv:2208.06193 (2022)
21. Wen, L., Duan, J., Li, S.E., Xu, S., Peng, H.: Safe reinforcement learning for autonomous vehicles through parallel constrained policy optimization. In: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC). pp. 1–7. IEEE (2020)
22. Yao, Y., Cen, Z., Ding, W., Lin, H., Liu, S., Zhang, T., Yu, W., Zhao, D.: Oasis: Conditional distribution shaping for offline safe reinforcement learning. Advances in Neural Information Processing Systems **37**, 78451–78478 (2024)
23. Zhang, L., Zhang, Q., Shen, L., Yuan, B., Wang, X., Tao, D.: Evaluating model-free reinforcement learning toward safety-critical tasks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 15313–15321 (2023)
24. Zheng, Y., Li, J., Yu, D., Yang, Y., Li, S.E., Zhan, X., Liu, J.: Safe offline reinforcement learning with feasibility-guided diffusion model. arXiv preprint arXiv:2401.10700 (2024)
25. Zhu, Z., Zhao, H., He, H., Zhong, Y., Zhang, S., Guo, H., Chen, T., Zhang, W.: Diffusion models for reinforcement learning: A survey. arXiv preprint arXiv:2311.01223 (2023)