SPOILDICE: SAFE AND PERFORMANT OFFLINE IMITATION LEARNING FROM DUAL DEMONSTRATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Ensuring both high returns and strict safety guarantees remains a fundamental challenge in imitation learning (IL). Existing approaches often rely on manually specified reward or cost functions, which are difficult to design and rarely capture complex safety constraints in real-world settings. We tackle this issue by introducing the problem of imitation learning from safety–performance dual demonstrations, where training data naturally divides into (i) safe demonstrations that respect safety requirements but may be suboptimal, and (ii) performant demonstrations that achieve high returns but may violate safety. To address the problem in the offline setting, we propose SpoilDICE (Safe–Performant Offline Imitation Learning via stationary DIstribution Correction Estimation). SpoilDICE integrates DICE-based distribution matching with support constraints derived from safe demonstrations, enabling agents to exploit high-return behaviors while remaining within safety-compliant regions of the state–action space. We validate our approach in both tabular gridworld and continuous safety-critical domains from the DSRL dataset and Safety Gymnasium benchmark. Empirical results demonstrate that SpoilDICE consistently produces policies that achieve strong performance without sacrificing safety, substantially outperforming prior offline IL baselines.

1 INTRODUCTION

Imitation learning (IL) is a powerful paradigm for addressing complex sequential decision-making tasks by directly leveraging expert demonstrations (Hussein et al., 2017; Ho & Ermon, 2016; Zare et al., 2024). In contrast to reinforcement learning (RL), which relies on carefully designed reward functions that are often difficult to specify and misaligned with the expert's true intent, IL offers a more practical alternative for real-world domains such as robotic manipulation (Fang et al., 2019; Xie et al., 2020; Johns, 2021) and autonomous driving (Le Mero et al., 2022; Hawke et al., 2020). By bypassing the challenge of reward engineering, IL enables agents to acquire sophisticated behaviors more directly from data.

Yet, applying IL in safety-critical domains introduces a new challenge: policies must not only perform well but also satisfy strict safety requirements. A typical way to enforce safety is through explicit cost functions and associated constraints (Yin et al., 2021; Ciftci et al., 2025; Shao et al., 2024), casting the problem as a constrained optimization. However, specifying such costs can be as difficult and error-prone as reward design itself, particularly when safety requirements are complex or task-dependent.

To mitigate this difficulty, inverse constraint learning (ICL) (Liu et al., 2025) has been proposed as a data-driven alternative that infers safety constraints directly from demonstrations labeled as safe. However, existing ICL methods, particularly those in the context of inverse constrained RL, typically assume access to a single type of safe demonstrations along with a predefined reward function (Malik et al., 2021; Hugessen et al., 2024). This assumption poses a substantial limitation, as specifying a reward function that faithfully captures the expert's intent is typically infeasible in real-world scenarios where IL is most needed. These challenges highlight the necessity of developing approaches that can directly learn policies satisfying both performance and safety objectives purely from demonstrations, without relying on explicit reward or cost signals.

This problem setting is inherently challenging, as disentangling performance and safety solely from a single expert demonstration is fundamentally difficult without additional prior knowledge. In prac-

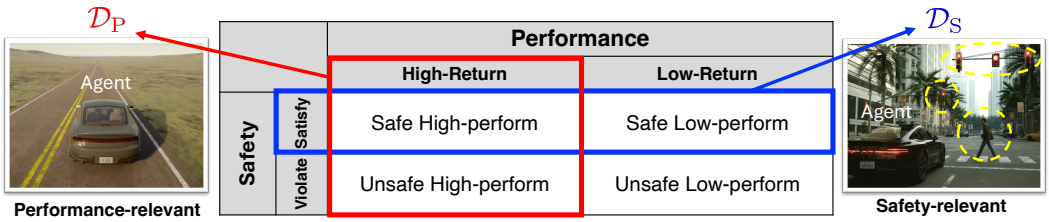


Figure 1: Illustrative example of safety–performance dual demonstrations. We consider two types of demonstrations, $\mathcal{D}_\mathrm{S}$ and $\mathcal{D}_\mathrm{P}$, where $\mathcal{D}_\mathrm{S}$ comprises constraint-abiding safe behaviors that may exhibit suboptimal performance, while $\mathcal{D}_\mathrm{P}$ consists of high performance behaviors that may violate underlying safety constraints.

tice, however, expert demonstrations often stratify naturally into distinct types (Kwon et al., 2024). For example, autonomous driving datasets may contain both aggressive, high-performance maneuvers and conservative, safety-focused behaviors (Bansal et al., 2019); robot manipulation tasks involving fragile objects exhibit demonstrations that prioritize either task completion speed or damage prevention (Mandlekar et al., 2021); and medical decision-making records capture both efficiency-driven and risk-averse treatment protocols (Nemati et al., 2016; Komorowski et al., 2018). Such natural stratification suggests that learning from explicit performance and safety demonstrations is often more viable than attempting to disentangle mixed objectives from homogeneous datasets.

While this dual demonstration provides a promising foundation for disentangling safety and performance, practical deployment scenarios further necessitate the offline setting. In many real-world domains, interactive data collection is costly, unsafe, or ethically restricted, making online interactions infeasible. Offline IL (Kim et al., 2022b;a; Xu et al., 2022; Yu et al., 2023) offers a pragmatic alternative by learning entirely from static demonstration datasets, which are readily available in areas such as autonomous driving logs, robotic teleoperation data, and clinical decision records. This offline regime not only reflects the practical constraints of data collection but also amplifies the importance of effectively leveraging heterogeneous demonstrations, as it can mitigate the need for corrective interaction during training. To this end, offline IL typically incorporates unlabeled demonstrations, which are substantially easier to collect at scale compared to expert demonstrations.

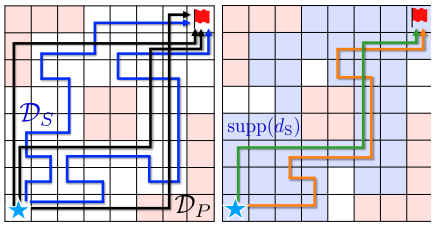
Building on these considerations, we introduce *offline imitation learning from safety–performance dual demonstrations (offline IL-SP)*. As illustrated in Figure 1, the dataset consists of **safe demonstrations** that satisfy safety constraints but may be suboptimal, **performant demonstrations** that achieve high returns but may violate safety. We then propose *SpoilDICE* (**S**afe-**p**erformant **o**ffline **i**mitation **l**earning via **Di**stribution **C**orrection **E**stimation), which augments the DICE framework with distributional support constraints induced by safe demonstrations to yield a constrained optimization problem. This formulation explicitly aligns policies with performant behaviors while enforcing safety guarantees, and further leverages unlabeled data to improve distribution estimation when labeled dual demonstrations are limited.

Our main contributions are summarized as follows:

- We introduce *imitation learning from safety–performance dual demonstrations (IL-SP)*, a previously unexplored setting where safe and performant demonstrations are provided as orthogonal learning signals. This formulation enables selective imitation of safe yet high-performing behaviors without any reward or cost specification.
- We extend IL-SP to the offline regime and propose SpoilDICE, the first DICE-based method that incorporates an explicit *support constraint* derived from safe demonstrations. This constraint fundamentally differs from conventional distribution-matching regularized DICE objectives and is crucial especially for IL-SP, to ignore suboptimal behaviors in safe demonstrations.
- We demonstrate effectiveness of our approach on an extensive set of benchmarks, including DSRL benchmarks (Liu et al., 2024) and Safety Gymnasium environments (Ji et al., 2023), showing consistent improvements over baseline methods while ensuring the safety constraints. Moreover, we empirically show that SpoilDICE is also efficient in terms of the number of trajectories and robust to the quality of safe and performance demonstrations respectively.

Performance Signal \ Safety Signal	No Explict Safety	Safety Constraint: Cost	Safety Constraint: Demonstration
Reward	OptiDICE (RL)	COptiDICE (Constrained RL)	ICSDICE (Inverse Constrained RL)
Demonstration	DemoDICE (IL)	– (Constrained IL)	**SpoilDICE** (IL-SP)

Distribution Constraint — Support Constraint



(a) Demonstrations (b) Constraint Functionality

Figure 2: (left) Taxonomy of DICE-based offline learning methods by available performance and safety signals. SpoilDICE uniquely applies an explicit *support constraint* from safe demonstrations while following performant behaviors. (right) Illustrative example contrasting *distribution constraint*, shown in orange, which matches suboptimal safe behaviors, and *support constraint*, shown in green, which restricts the policy to the safe support, regardless of its value.

2 Related Work

Safe Imitation Learning One line of research incorporates safety into IL by imposing cost or constraint signals. Recent methods address cost-constrained behaviors by optimizing policies under trajectory cost upper bounds (Shao et al., 2024), while Cao et al. (2025) propose a simple approach to constraint-aware imitation learning that integrates a differentiable safety filter into the policy. SAFE-GIL (Ciftci et al., 2025) introduces adversarial perturbations during online demonstration collection to elicit recovery behaviors in safety-critical states. Notably, SafeDICE (Jang et al., 2023) enforces constraint satisfaction from non-preferred demonstrations without using any explicit cost function. Although these approaches incorporate safety into imitation, they often rely on an explicitly predefined cost signal or operate along a single constraint axis, whereas our offline IL-SP explicitly separates safety and performance as orthogonal dimensions.

Inverse Constrained Learning ICL seeks to infer latent cost or constraint functions from demonstrations (Liu et al., 2025). Its extension to inverse constrained RL (ICRL) combines such inferred constraints with policy optimization (Malik et al., 2021; Kim et al., 2023; Subramanian et al., 2024; Hugessen et al., 2024), but typically assumes access to explicit rewards or allows online interaction. More recent work explores an offline variant of ICRL (Quan et al., 2024), yet these approaches still focus on inferring hidden cost functions, while our method directly leverages the explicit separation between safe and performant demonstrations as learning signals.

Offline Imitation Learning Offline IL is crucial for domains where interactive data collection is costly, unsafe, or ethically restricted, yet offline datasets often contain demonstrations of mixed quality. To handle this, DWBC reweights behavioral cloning via a discriminator (Xu et al., 2022), and SPRINQL penalizes undesirable behaviors by emphasizing expert-consistent trajectories (Hoang et al., 2024). Other approaches leverage transition dynamics through DICE: DemoDICE extends stationary distribution matching to imperfect data (Kim et al., 2022b), while RelaxDICE relaxes divergence penalties to implicitly regularize support (Yu et al., 2023). These methods mitigate issues of mixed-quality data but do not disentangle safety and performance, which is central to our offline IL-SP formulation. We summarize the taxonomy of DICE-based approaches in Figure 2. For more detailed discussion on offline learning method, see Section B.

Novelty of SpoilDICE Prior DICE-based offline IL methods fundamentally rely on *distribution constraints* that regularize the stationary distribution toward the behavior distribution. When directly applied to the IL–SP setting (Figure 2), this regularization keeps the policy inside the safe data distribution but simultaneously pulls it toward state–action pairs that occur frequently in the safe demonstrations, including suboptimal safe behaviors. This phenomenon is clearly illustrated in the gridworld example, where the distribution constraint biases the policy toward frequently visited but suboptimal safe regions (orange path). In contrast, a *support constraint* does not match densities. It restricts the policy only to the safe support while still allowing alignment with performant behaviors, enabling recovery of both safety and performance (green path). Moreover, when safe and unlabeled demonstrations coexist, the support constraint offers a principled advantage: it enforces safety via the empirically identified safe region while using mixed data to infer performant behavior. To the best of our knowledge, SpoilDICE is the first DICE-based offline IL method that explicitly incorporates and operationalizes such a *safe-support* constraint

3 PRELIMINARIES

3.1 CONSTRAINED MARKOV DECISION PROCESS

We consider a Markov decision process (MDP) defined as a tuple $\mathcal{M} := \langle \mathcal{S}, \mathcal{A}, T, \rho_0, r, \gamma \rangle$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces respectively, $T : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition probability function, $\rho_0 \in \Delta(\mathcal{S})$ is the initial state distribution, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, and $\gamma \in (0, 1)$ is the discount factor. We define the policy space as $\Pi := \{\pi : \mathcal{S} \to \Delta(\mathcal{A})\}$.

For a given policy $\pi \in \Pi$, the stationary distribution d_π is defined as:

$$d_\pi(s, a) := (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s, a_t = a),$$

where $s_0 \sim \rho_0$, $a_t \sim \pi(s_t)$, and $s_{t+1} \sim T(s_t, a_t)$. Then, d_π satisfies the Bellman flow constraint:

$$\sum_{a \in \mathcal{A}} d_\pi(s, a) = (1 - \gamma)\rho_0(s) + \gamma \sum_{\bar{s}, \bar{a}} T(s|\bar{s}, \bar{a}) d_\pi(\bar{s}, \bar{a}), \quad \forall s \in \mathcal{S}.$$

Conversely, from a distribution d satisfying the Bellman flow constraint, we can construct a policy π:

$$\pi(a|s) := \frac{d_\pi(s, a)}{\sum_{\bar{a} \in \mathcal{A}} d_\pi(s, \bar{a})}, \quad \forall s \in \mathcal{S}.$$

For any function $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, we denote the expected value under the stationary distribution as:

$$J(\pi; f) := \mathbb{E}_{(s,a) \sim d_\pi}[f(s, a)].$$

A constrained Markov decision process (CMDP) (Altman, 2021) extends the standard MDP framework to incorporate safety constraints. We define a CMDP as $\mathcal{M}_c := \langle \mathcal{M}, c \rangle$, where $\mathcal{M}$ is the underlying MDP and $c : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_{\geq 0}$ is a non-negative cost function, where both the cost function c and the reward function r are unobservable to the agent in our problem setting.

3.2 IMITATION LEARNING

Imitation learning (IL) aims to recover an expert policy from demonstrations in the absence of reward signals. We denote the demonstration dataset as $\mathcal{D} = \{(s, a)\}$, a collection of state-action pairs sampled from expert trajectories. A straightforward approach to IL is behavioral cloning (BC), which treats the problem as supervised learning and solves it by minimizing the negative log-likelihood:

$$\min_{\pi} \quad \mathbb{E}_{(s,a) \sim \mathcal{D}} \left[-\log \pi(a|s) \right].$$

However, BC suffers from compounding errors due to covariate shift, particularly over long horizons (Ross & Bagnell, 2010; Seo et al., 2024). To address this limitation, recent IL algorithms typically minimize a distribution matching objective:

$$\min_{\pi} \quad \mathbb{D}(d_\pi(s, a), d_{\mathcal{D}}(s, a)), \tag{1}$$

where $\mathbb{D}$ is a divergence measure (e.g., f-divergences (Kim et al., 2022b; Zhang et al., 2020; Ho & Ermon, 2016), integral probability metrics (Kim & Park, 2018; Dadashi et al., 2020)) and $d_{\mathcal{D}}$ denotes the state-action distribution in $\mathcal{D}$.

4 OFFLINE IL WITH SAFETY–PERFORMANCE DEMONSTRATIONS

In this section, we formally define the problem of offline IL from safety–performance dual demonstrations (offline IL-SP), where environment interactions are prohibited during training.

4.1 IMITATION LEARNING FROM SAFETY–PERFORMANCE DUAL DEMONSTRATIONS

We consider imitation learning from **safety–performance dual demonstrations** (IL-SP), consisting of a set of *safe* demonstrations $\mathcal{D}_{\mathrm{S}}$, which satisfy the underlying constraints but may be suboptimal, and a set of *performant* demonstrations $\mathcal{D}_{\mathrm{P}}$, which achieve high returns but may violate unknown safety constraints (Figure 1). Denote state-action distributions of $\mathcal{D}_{\mathrm{S}}, \mathcal{D}_{\mathrm{P}}$ as $d_{\mathrm{S}}, d_{\mathrm{P}}$, respectively.

Since $\mathcal{D}_{\mathrm{S}}$ may include suboptimal trajectories, our objective is to capture only the safety-compliant support region rather than to match its distribution d_{S}. Accordingly, we require the policy π to align with performant behaviors while remaining restricted to the safe support. Hence, the goal of IL-SP is to recover a policy π whose stationary distribution d_π matches the performant distribution d_{P} while lying within the support of the safe distribution d_{S}:

$$\min_{\pi} \quad D_{\mathrm{KL}}(d_\pi \| d_{\mathrm{P}}) \quad \text{subject to} \quad \mathrm{supp}(d_\pi) \subseteq \mathrm{supp}(d_{\mathrm{S}}), \tag{2}$$

where $\mathrm{supp}(d) = \{(s,a) \mid d(s,a) > 0\}$. In particular, IL-SP reduces to the following cost-constrained IL formulation,

$$\min_{\pi} \quad D_{\mathrm{KL}}(d_\pi \,\|\, d_{\mathrm{P}}) \quad \text{subject to} \quad J(\pi; c) = 0,$$

when the cost function is non-negative ($c \geq 0$). This follows because any probability mass placed outside the safe support necessarily yields $J(\pi; c) > 0$. Thus, IL-SP can be understood as a cost-constrained imitation learning problem derived directly from dual demonstrations, without the need for explicitly defined cost functions. A more detailed discussion is provided in Section A.

4.2 Offline IL from Dual Demonstrations with Support Constraint

We study the offline version of the IL-SP setting, which is particularly relevant for real-world applications where interaction with the environment is prohibited during training, and only static demonstrations are available. Following standard practice in offline IL (discussed in Section 3.2), we additionally assume access to an unlabeled dataset $\mathcal{D}_{\mathrm{U}}$ with state–action distribution d_{U}, which provides transition information without safety or performance labels.

Since direct sampling from d_π is infeasible in the offline setting, we instead optimize over a stationary distribution d corresponding to d_π, following DICE-based approaches for offline learning (Lee et al., 2021; 2022; Kim et al., 2022b). To enforce the support constraint $\mathrm{supp}(d_\pi) \subseteq \mathrm{supp}(d_{\mathrm{S}})$ from objective (2), we reformulate it as an equality condition:

$$\mathbb{E}_{(s,a)\sim d_{\mathrm{S}}}\left[\frac{d(s,a)}{d_{\mathrm{S}}(s,a)}\right] = 1 \quad \Longleftrightarrow \quad \sum_{s,a} \mathbb{1}_{d_{\mathrm{S}}}(s,a) d(s,a) = 1, \tag{3}$$

where $\mathbb{1}_{d_{\mathrm{S}}}(s,a) = 1$ if $(s,a) \in \mathrm{supp}(d_{\mathrm{S}})$ and 0 otherwise.

Then, our objective is to find a stationary distribution d that minimizes divergence to d_{P} while satisfying the support constraint with respect to d_{S}. To ensure d corresponds to a valid policy, we impose Bellman flow constraints and arrive at the following optimization problem:

$$\begin{aligned}
\min_{d} \quad & D_{\mathrm{KL}}(d \| d_{\mathrm{P}}) \\
\text{s.t.} \quad & \sum_{s,a} \mathbb{1}_{d_{\mathrm{S}}}(s,a) d(s,a) = 1 \\
& \sum_{a} d(s,a) = (1-\gamma)\rho_0(s) + \gamma \sum_{\bar{s},\bar{a}} T(s|\bar{s},\bar{a}) d(\bar{s},\bar{a}), \quad \forall s \in \mathcal{S} \\
& d(s,a) \geq 0, \quad \forall s \in \mathcal{S}, a \in \mathcal{A}.
\end{aligned} \tag{4}$$

Introducing Lagrange multipliers $\lambda \in \mathbb{R}$ for the support constraint and $\nu(s) \in \mathbb{R}$ for the Bellman flow constraints, we obtain the unconstrained objective:

$$\begin{aligned}
\min_{d\geq 0} \max_{\nu} \max_{\lambda} \quad & D_{\mathrm{KL}}(d \| d_{\mathrm{P}}) + \lambda \left(1 - \sum_{s,a} \mathbb{1}_{d_{\mathrm{S}}}(s,a) d(s,a)\right) \\
& + \sum_{s\in\mathcal{S}} \nu(s) \left(\sum_{a} d(s,a) - (1-\gamma)\rho_0(s) - \gamma \sum_{\bar{s},\bar{a}} T(s|\bar{s},\bar{a}) d(\bar{s},\bar{a})\right).
\end{aligned} \tag{5}$$

By exchanging the sign of optimization and then simplifying the result, we get:

$$\begin{aligned}
\max_{d\geq 0} \min_{\nu} \min_{\lambda} \quad & (1-\gamma)\mathbb{E}_{s\sim\rho_0}[\nu(s)] - \lambda \\
& + \mathbb{E}_{(s,a)\sim d}\left[\gamma(T\nu)(s,a) - \nu(s) - \log \frac{d(s,a)}{d_{\mathrm{P}}(s,a)} + \lambda \mathbb{1}_{d_{\mathrm{S}}}(s,a)\right],
\end{aligned} \tag{6}$$

where $(T\nu)(s,a) := \mathbb{E}_{s' \sim T(s,a)}[\nu(s')]$. To leverage $\mathcal{D}_\mathrm{U}$, we apply the change of variables $w(s,a) := \frac{d(s,a)}{d_\mathrm{U}(s,a)}$ and define the density ratio $r_\mathrm{P}(s,a) := \frac{d_\mathrm{P}(s,a)}{d_\mathrm{U}(s,a)}$. This transformation yields:

$$
\begin{aligned}
&(1-\gamma)\mathbb{E}_{s\sim\rho_0}[\nu(s)] - \lambda \\
&+ \mathbb{E}_{(s,a)\sim d_\mathrm{U}}\Big[w(s,a)\big(\underbrace{\gamma(T\nu)(s,a) - \nu(s) + \log r_\mathrm{P}(s,a) + \lambda \mathbb{1}_{d_\mathrm{S}}(s,a)}_{=:A_{\nu,\lambda}(s,a)} - \log w(s,a)\big)\Big] \qquad (7)\\
&= (1-\gamma)\mathbb{E}_{s\sim\rho_0}[\nu(s)] + \mathbb{E}_{(s,a)\sim d_\mathrm{U}}\left[w(s,a)(A_{\nu,\lambda}(s,a) - \log w(s,a))\right] - \lambda =: L(w,\nu,\lambda) \qquad (8)
\end{aligned}
$$

Then, our optimization objective becomes:

$$\min_{\nu}\min_{\lambda}\max_{w\geq 0} L(w,\nu,\lambda). \qquad (9)$$

Proposition 1. *(Strong Duality) $L(w,\nu,\lambda)$ satisfies strong duality, which ensures:*

$$\max_{w\geq 0}\min_{\nu,\lambda} L(w,\nu,\lambda) = \min_{\nu,\lambda}\max_{w\geq 0} L(w,\nu,\lambda). \qquad (10)$$

Proposition 2. *(Closed-form solution; Lee et al. (2022)) The inner maximization on the RHS of Eq. (10) admits a closed-form solution:*

$$w^*_{\nu,\lambda}(s,a) = \exp(A_{\nu,\lambda}(s,a) - 1). \qquad (11)$$

See Section C for the proofs of these propositions. Hence, the problem is converted into the following minimization problem w.r.t ν, λ:

$$
\begin{aligned}
&\min_{\nu}\min_{\lambda} \quad (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a)\sim d_\mathrm{U}}\left[w^*(s,a)(A_{\nu,\lambda}(s,a) - \log w^*(s,a))\right] - \lambda \\
&= \min_{\nu}\min_{\lambda} \quad (1-\gamma)\mathbb{E}_{s\sim p_0}[\nu(s)] + \mathbb{E}_{(s,a)\sim d_\mathrm{U}}[\exp(A_{\nu,\lambda}(s,a) - 1)] - \lambda =: L(\nu,\lambda). \qquad (12)
\end{aligned}
$$

We refer to the method that optimizes the objective in Equation (12) as SpoilDICE, short for **S**afe–**p**erformant **o**ffline **i**mitation **l**earning via stationary **DI**stribution **C**orrection **E**stimation.

4.3 Support Estimation with Unlabeled Demonstrations

To optimize the objective in Equation (12), we need to estimate the support indicator $\mathbb{1}_{d_\mathrm{S}}$ from demonstrations, since the density $d_\mathrm{S}(s,a)$ is not directly accessible. For this, we adopt the results from PU-learning (Scott & Blanchard, 2009) to estimate the support indicator, treating $\mathcal{D}_\mathrm{U}$ as unlabeled data and $\mathcal{D}_\mathrm{S}$ as positive-labeled data. Intuitively, the unlabeled-date distribution $p(x)$ is just a mixture of positive examples $p(x|y=1)$ and negative examples $p(x|y=0)$. Formally, if the fraction of positives is $\eta = p(y=1)$, then

$$p(x) = \eta\, p(x|y=1) + (1-\eta)\, p(x|y=0).$$

Rearranging the terms yields

$$\underbrace{\frac{p(x|y=0)}{p(x|y=1)}}_{=\text{unsafe-safe ratio}} = \frac{1}{(1-\eta)}\ \underbrace{\frac{p(x)}{p(x|y=1)}}_{=\text{unlabeled-safe ratio}} - \frac{\eta}{1-\eta},$$

showing that the unsafe–safe ratio can be obtained from the unlabeled–safe ratio, which can be estimated from $\mathcal{D}_\mathrm{U}$ and $\mathcal{D}_\mathrm{S}$ in our setting, e.g. using generative adversarial training (Goodfellow et al., 2014). This formulation exploits unlabeled data to capture negative information, and is therefore more robust than one-class classification methods (Schölkopf et al., 2001; Steinwart et al., 2005), which often fail when the negative distribution changes.

Following this approach, we train discriminator $\hat{C}_\mathrm{S} : \mathcal{S}\times\mathcal{A} \to [0,1]$ to estimate the density ratio $\hat{r}_\mathrm{S} \approx d_\mathrm{S}/d_\mathrm{U}$ for $\mathbb{1}_{d_\mathrm{S}}$, and similarly discriminator $\hat{C}_\mathrm{P}$ to estimate $\hat{r}_\mathrm{P} \approx d_\mathrm{P}/d_\mathrm{U}$ for r_P appearing in $A_{\nu,\lambda}$. See Section D.1 for the details.

To construct a support estimator $\hat{\mathbb{1}}_{d_S}$, we set a quantile cutoff κ (e.g., the 0.01-quantile) that determines the decision threshold of $\hat{r}_S(s,a)$ evaluated on $\mathcal{D}_S$. This empirical choice retains nearly all samples from $\mathcal{D}_S$, while treating only the extreme lower tail as out-of-support, thereby calibrating the estimator conservatively. The resulting support estimator is defined as follows:

$$\hat{\mathbb{1}}_{d_S}(s,a) = \begin{cases} 1, & \text{if } \hat{r}_S(s,a) \geq \text{Quantile}_\kappa(\hat{r}_S, \mathcal{D}_S), \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

We then incorporate $\hat{r}_P$ and $\hat{\mathbb{1}}_{d_S}$ into $L_{\nu,\lambda}$ in Eq. (12) and jointly optimize ν and λ to obtain the optimal weight $w^*_{\nu,\lambda}$ using a closed-form solution in Eq. (11). Following prior DICE approaches (Lee et al., 2022; Kim et al., 2022b), the final policy π^* is extracted from $w^*_{\nu,\lambda}$ via weighted BC (see Section D.3). A full description of the SpoilDICE algorithm is provided in Section D.4.

5 Experiments

We evaluate our approach in two settings: (1) tabular gridworld experiments with discrete state–action spaces and (2) realistic tasks from DSRL Liu et al. (2024) and Safety Gymnasium (Ji et al., 2023) benchmarks. As the problem setting is novel, no direct baselines exist. We therefore evaluate SpoilDICE against the following adapted baselines.

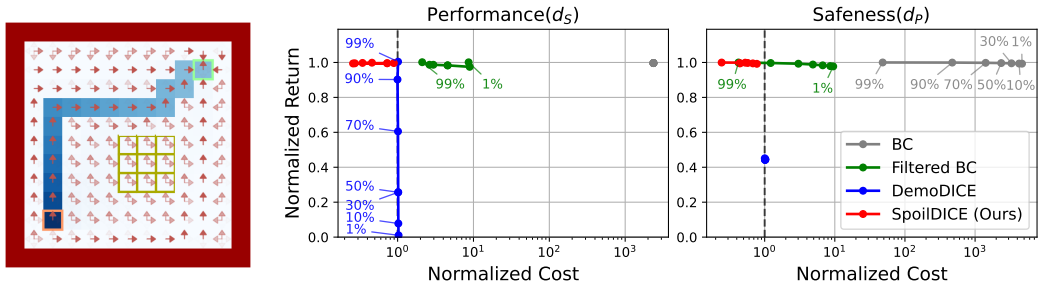
- **BC**: Standard behavioral cloning, minimizing $\mathbb{E}_{(s,a)\sim d_P}[-\log \pi(a|s)]$.
- **Filtered BC (FBC)**: BC restricted to samples retained by the estimated support $\hat{\mathbb{1}}_{d_S}$, $\mathbb{E}_{(s,a)\sim d_P}[-\hat{\mathbb{1}}_{d_S}(s,a) \log \pi(a|s)]$.
- **Jointly Filtered BC (JFBC)**: BC applied only to samples accepted by both $\hat{\mathbb{1}}_{d_S}$ and $\hat{\mathbb{1}}_{d_P}$, $\mathbb{E}_{(s,a)\sim d_U}[-\hat{\mathbb{1}}_{d_S}(s,a)\hat{\mathbb{1}}_{d_P}(s,a) \log \pi(a|s)]$.
- **DemoDICE-S**, **DemoDICE-U**: Variants of DemoDICE (Kim et al., 2022b) that regularize d toward d_S, differing in the choice of importance sampling distribution (d_S or d_U respectively).
- **DWBC-S**: Variants of DWBC (Xu et al., 2022) that aims to recover performant behaviors from safe demonstrations by training a discriminator based on PU-learning objective.
- **SafeDICE** (Jang et al., 2023): DICE-based IL that uses non-preferred and unlabeled demonstrations. For conceptual comparison, we treat unsafe demonstrations as non-preferred ones.
- **COptiDICE** (Lee et al., 2022): DICE-based constrained RL. For conceptual comparison, we use the true rewards and costs provided in the dataset.

5.1 Tabular Experiments: Gridworld with Constraint Zones

In the tabular experiment, our goal is to examine how each original objective behaves under different conditions of d_S and d_P. To this end, we design a constrained gridworld environment with randomly placed unsafe zones, as shown in Figure 3a. The task is to navigate from the start position (orange) to the goal (green) as quickly as possible while avoiding 3×3 unsafe zones (olive) arranged along the main diagonal. This synthetic domain allows exact evaluation of distribution-level objectives, since the true distributions d_P and d_S can be computed from the transition model, enabling exact solutions for each method.

To evaluate the robustness of each method, we systematically vary the qualities of d_P and d_S, and report normalized return and episode cost averaged over 50 runs. Here, the **safeness** of d_P measures how reliably it avoids unsafe states, while the **performance** of d_S measures how well it achieves high return.

Figure 3b summarizes the results. Filtered BC suffers higher costs as data quality decreases, since it depends only on the intersection of d_P and d_S, which rarely retains safe–performant behaviors under poor-quality demonstrations. DemoDICE performs well only when d_S is near-optimal, otherwise inheriting its suboptimality. In contrast, SpoilDICE consistently balances return and safety by aligning with d_P while enforcing support constraints from d_S, demonstrating robustness across all tested conditions (see Section E for details).



(a) Constrained gridworld.

(b) Evaluation under varying qualities of $d_{\mathrm{S}}, d_{\mathrm{P}}$.

Figure 3: Description of the tabular experiment. (a) Optimal behavior in the constrained gridworld. (b) Safety–performance trade-off under varying qualities of d_{S} (performance) and d_{P} (safeness). Each point corresponds to a policy learned by one method, evaluated in terms of normalized return (y-axis) and normalized cost (x-axis). The labels (e.g., "99%", "30%") indicate the proportion of high-quality demonstrations used to construct d_{S} or d_{P}. SpoilDICE remains near the top-left region across all settings, showing robustness to both low-quality d_{P} (left) and d_{S} (right).

5.2 Realistic Experiments: DSRL and Safety Gymnasium Benchmark

To assess our approach in more realistic settings, we conduct continuous control experiments on the public benchmarks, including DSRL (Datasets for Safe RL) dataset (Liu et al., 2024) and Safety Gymnasium benchmark (Ji et al., 2023). We focus on the following research questions:

- **Q1**: Are the methods effective under realistic data collection scenarios?
- **Q2**: How data-efficient is each method with respect to the sizes of $\mathcal{D}_{\mathrm{S}}$ and $\mathcal{D}_{\mathrm{P}}$?
- **Q3**: How robust is each method to variations in data quality within $\mathcal{D}_{\mathrm{S}}$ and $\mathcal{D}_{\mathrm{P}}$?

To address these questions, we design two distinct data configurations. First, to evaluate feasibility in a realistic setting, we use the public DSRL dataset (Liu et al., 2024), which provides demonstrations covering a broad range of performance and safety levels. Second, to analyze robustness, we construct what we term *controlled datasets*, in which demonstrations are systematically composed from four distinct types to allow precise variation in safety and performance quality.

For evaluation, we use two metrics. (1) **Feasible return**: the cumulative reward up to the first cost violation in an episode, following Liu et al. (2023). (2) **Episode cost**: the total cumulative cost in each episode. All results are averaged over the last 10 evaluations across multiple random seeds.

5.2.1 Effectiveness on DSRL Dataset

To address **Q1**, we evaluate all methods on the 21 tasks provided in the DSRL benchmark. To utilize the offline dataset suitable to offline IL-SP problem, we label each trajectory as performant or non-performant, and as safe or unsafe, according to predefined return and cost thresholds consistent with the IL-SP setting (see Section F.1). Using these labels, we construct $\mathcal{D}_{\mathrm{P}}$ and $\mathcal{D}_{\mathrm{S}}$. In experiments, we use 10 trajectories randomly taking out from the $\mathcal{D}_{\mathrm{P}}$ and $\mathcal{D}_{\mathrm{S}}$ respectively, while $\mathcal{D}_{\mathrm{U}}$ consists of the entire offline dataset. For SafeDICE, we use the entire set of unsafe demonstrations as non-preferred.

The results are summarized in Table 1. Notably, SpoilDICE satisfies the safety constraint on all tasks we considered and outperforms strong baselines on 16 out of the 21 tasks with ensuring the safety conditions. It is also interesting that SpoilDICE achieves superior performance even compared to COptiDICE, an offline constrained RL method that has access to both explicit reward and cost signals. As aligned with the discussion in the DSRL paper (Liu et al., 2024), COptiDICE relies on a behavior-distribution constraint that induces conservatism, which can lead to suboptimality in both safety and performance depending on the dataset quality. In contrast, SpoilDICE leverages a support constraint, making it considerably less sensitive to the quality of safe demonstrations.

5.2.2 Data Efficiency

Regarding **Q2**, we examine the data efficiency of each method with respect to the sizes of $\mathcal{D}_{\mathrm{P}}$ and $\mathcal{D}_{\mathrm{S}}$. DSRL datasets are unsuitable for this analysis, as they inherently mix demonstrations of varying quality, making it difficult to isolate dataset size effects. To address this, we construct what we term *controlled datasets*, composed of four demonstration types as illustrated in Figure 1.

Task	Metric	BC	FBC	JFBC	DemoDICE-S	DemoDICE-U	DWBC-S	SafeDICE	COptiDICE	SpoilDICE (Ours)
PointCircle1	Feasible return	0.28 ± 0.01	0.07 ± 0.05	0.23 ± 0.00	**0.37 ± 0.03**	0.23 ± 0.02	0.33 ± 0.04	0.23 ± 0.02	0.22 ± 0.02	**0.47 ± 0.02**
	Episode cost	4.37 ± 0.15	5.10 ± 0.84	4.73 ± 0.06	**0.93 ± 0.38**	4.74 ± 0.20	3.49 ± 0.46	4.02 ± 0.29	4.57 ± 0.28	**0.27 ± 0.07**
PointCircle2	Feasible return	0.39 ± 0.04	0.12 ± 0.12	0.43 ± 0.01	**0.64 ± 0.00**	0.43 ± 0.00	0.40 ± 0.02	0.41 ± 0.00	0.46 ± 0.02	**0.66 ± 0.04**
	Episode cost	2.77 ± 0.11	1.93 ± 0.58	2.94 ± 0.05	**0.85 ± 0.09**	2.94 ± 0.05	2.64 ± 0.27	2.63 ± 0.13	2.24 ± 0.21	**0.93 ± 0.19**
PointButton1	Feasible return	0.26 ± 0.01	0.03 ± 0.02	0.28 ± 0.01	**0.03 ± 0.01**	**0.06 ± 0.02**	0.23 ± 0.01	0.08 ± 0.01	**0.09 ± 0.01**	**0.10 ± 0.02**
	Episode cost	2.93 ± 0.09	1.81 ± 0.28	3.23 ± 0.10	**0.51 ± 0.07**	**0.95 ± 0.14**	3.02 ± 0.10	1.04 ± 0.14	**0.92 ± 0.12**	**0.95 ± 0.09**
PointButton2	Feasible return	0.28 ± 0.02	0.12 ± 0.02	0.38 ± 0.01	**0.17 ± 0.02**	**0.19 ± 0.02**	0.29 ± 0.01	**0.20 ± 0.02**	**0.18 ± 0.02**	**0.21 ± 0.01**
	Episode cost	1.58 ± 0.04	1.90 ± 0.09	1.70 ± 0.06	**0.69 ± 0.06**	**0.73 ± 0.06**	1.60 ± 0.08	**0.76 ± 0.06**	**0.67 ± 0.03**	**0.92 ± 0.08**
PointGoal1	Feasible return	**0.45 ± 0.02**	0.14 ± 0.02	0.57 ± 0.01	**0.23 ± 0.02**	**0.55 ± 0.01**	**0.47 ± 0.01**	**0.53 ± 0.02**	**0.56 ± 0.01**	**0.34 ± 0.02**
	Episode cost	**0.92 ± 0.03**	1.15 ± 0.16	1.00 ± 0.04	**0.35 ± 0.04**	**0.88 ± 0.04**	**0.94 ± 0.04**	**0.97 ± 0.05**	**0.88 ± 0.04**	**0.46 ± 0.02**
PointGoal2	Feasible return	0.41 ± 0.01	0.12 ± 0.01	0.49 ± 0.01	**0.17 ± 0.01**	0.43 ± 0.01	0.37 ± 0.01	0.45 ± 0.01	**0.41 ± 0.01**	**0.32 ± 0.02**
	Episode cost	1.16 ± 0.09	1.30 ± 0.10	1.35 ± 0.01	**0.40 ± 0.04**	1.14 ± 0.05	1.19 ± 0.05	1.04 ± 0.06	**0.99 ± 0.05**	**0.61 ± 0.05**
HopperVelocity	Feasible return	0.18 ± 0.04	**0.84 ± 0.00**	0.27 ± 0.04	**0.09 ± 0.02**	0.16 ± 0.05	0.17 ± 0.03	**0.23 ± 0.07**	0.21 ± 0.10	**0.48 ± 0.07**
	Episode cost	1.71 ± 0.32	**0.00 ± 0.00**	2.97 ± 0.60	**0.52 ± 0.23**	2.39 ± 0.96	2.28 ± 0.21	**0.98 ± 0.31**	1.16 ± 0.45	**0.60 ± 0.12**
HalfCheetahVelocity	Feasible return	0.67 ± 0.09	**0.66 ± 0.07**	0.54 ± 0.11	**0.71 ± 0.03**	0.39 ± 0.05	0.72 ± 0.06	**0.67 ± 0.10**	**0.73 ± 0.05**	**0.75 ± 0.02**
	Episode cost	1.92 ± 0.41	**0.13 ± 0.06**	2.82 ± 0.73	**0.74 ± 0.14**	4.30 ± 0.80	1.44 ± 0.28	**0.82 ± 0.12**	**0.78 ± 0.20**	**0.45 ± 0.12**
Walker2dVelocity	Feasible return	0.22 ± 0.01	**0.25 ± 0.04**	0.23 ± 0.01	**0.67 ± 0.04**	0.46 ± 0.04	0.28 ± 0.01	**0.74 ± 0.04**	**0.71 ± 0.04**	**0.79 ± 0.01**
	Episode cost	3.27 ± 0.07	**0.49 ± 0.06**	3.35 ± 0.04	**0.94 ± 0.26**	2.25 ± 0.21	2.96 ± 0.09	**0.40 ± 0.21**	**0.69 ± 0.30**	**0.07 ± 0.02**
AntVelocity	Feasible return	0.78 ± 0.08	**−0.02 ± 0.01**	0.78 ± 0.05	**0.96 ± 0.00**	0.57 ± 0.09	0.75 ± 0.05	0.40 ± 0.07	0.31 ± 0.02	**0.96 ± 0.00**
	Episode cost	1.34 ± 0.22	**0.04 ± 0.01**	1.31 ± 0.15	**0.30 ± 0.05**	2.18 ± 0.38	1.33 ± 0.13	3.53 ± 0.66	3.93 ± 0.35	**0.15 ± 0.03**
BallRun	Feasible return	0.33 ± 0.04	0.35 ± 0.08	0.37 ± 0.02	0.27 ± 0.05	**0.01 ± 0.00**	0.39 ± 0.02	0.44 ± 0.01	0.43 ± 0.00	**0.36 ± 0.02**
	Episode cost	1.18 ± 0.16	1.01 ± 0.05	1.61 ± 0.15	1.36 ± 0.38	**0.00 ± 0.00**	1.14 ± 0.11	1.17 ± 0.05	1.40 ± 0.04	**0.86 ± 0.19**
BallCircle	Feasible return	0.64 ± 0.01	**0.77 ± 0.03**	0.65 ± 0.00	**0.55 ± 0.02**	0.68 ± 0.01	0.66 ± 0.02	**0.65 ± 0.01**	0.68 ± 0.01	**0.65 ± 0.01**
	Episode cost	1.44 ± 0.02	**0.94 ± 0.04**	1.44 ± 0.03	**0.48 ± 0.05**	1.34 ± 0.02	1.23 ± 0.04	**0.94 ± 0.05**	1.00 ± 0.05	**0.54 ± 0.05**
CarRun	Feasible return	**0.95 ± 0.02**	**0.96 ± 0.00**	**0.97 ± 0.01**	0.59 ± 0.03	**0.48 ± 0.07**	**0.99 ± 0.00**	**0.98 ± 0.00**	**0.98 ± 0.00**	**0.97 ± 0.00**
	Episode cost	**0.45 ± 0.15**	**0.51 ± 0.04**	**0.37 ± 0.05**	1.79 ± 0.58	**0.42 ± 0.32**	**0.16 ± 0.02**	**0.01 ± 0.01**	**0.02 ± 0.01**	**0.07 ± 0.03**
CarCircle	Feasible return	0.38 ± 0.02	0.52 ± 0.03	0.40 ± 0.00	**0.54 ± 0.03**	0.42 ± 0.01	0.44 ± 0.02	0.47 ± 0.02	0.44 ± 0.01	**0.64 ± 0.01**
	Episode cost	2.27 ± 0.06	1.20 ± 0.08	2.19 ± 0.04	**0.52 ± 0.16**	2.15 ± 0.02	1.85 ± 0.08	1.40 ± 0.04	1.57 ± 0.06	**0.82 ± 0.03**
DroneRun	Feasible return	0.29 ± 0.04	0.33 ± 0.02	0.33 ± 0.04	0.34 ± 0.02	0.42 ± 0.01	0.31 ± 0.01	0.43 ± 0.01	0.44 ± 0.03	**0.58 ± 0.01**
	Episode cost	2.34 ± 0.16	2.57 ± 0.21	2.41 ± 0.36	2.55 ± 0.14	2.15 ± 0.06	2.67 ± 0.13	1.49 ± 0.12	1.82 ± 0.07	**0.68 ± 0.20**
DroneCircle	Feasible return	0.16 ± 0.01	0.09 ± 0.01	0.35 ± 0.02	**0.33 ± 0.01**	0.54 ± 0.01	0.27 ± 0.03	0.66 ± 0.01	0.69 ± 0.01	**0.62 ± 0.02**
	Episode cost	1.25 ± 0.13	1.12 ± 0.06	1.77 ± 0.12	**0.81 ± 0.06**	1.61 ± 0.04	1.61 ± 0.09	1.09 ± 0.08	1.06 ± 0.04	**0.80 ± 0.04**
AntRun	Feasible return	0.34 ± 0.00	**0.10 ± 0.01**	0.40 ± 0.01	**0.41 ± 0.03**	0.50 ± 0.03	0.35 ± 0.01	0.62 ± 0.02	0.48 ± 0.01	**0.53 ± 0.01**
	Episode cost	1.60 ± 0.12	**0.02 ± 0.00**	1.99 ± 0.17	**0.19 ± 0.04**	1.86 ± 0.18	1.68 ± 0.14	1.20 ± 0.11	1.88 ± 0.04	**0.46 ± 0.13**
AntCircle	Feasible return	0.13 ± 0.01	**0.08 ± 0.01**	0.33 ± 0.00	**0.17 ± 0.01**	0.34 ± 0.01	**0.15 ± 0.01**	0.32 ± 0.00	0.32 ± 0.02	**0.33 ± 0.01**
	Episode cost	1.27 ± 0.10	**0.47 ± 0.10**	1.22 ± 0.08	**0.49 ± 0.05**	1.77 ± 0.07	**0.94 ± 0.05**	2.05 ± 0.12	2.05 ± 0.08	**0.48 ± 0.06**
MD-easydense	Feasible return	0.50 ± 0.01	**0.50 ± 0.02**	0.45 ± 0.01	**0.47 ± 0.09**	0.48 ± 0.01	0.51 ± 0.02	0.53 ± 0.01	**0.45 ± 0.03**	**0.58 ± 0.03**
	Episode cost	1.60 ± 0.02	**0.49 ± 0.02**	1.35 ± 0.04	**0.31 ± 0.16**	1.45 ± 0.03	1.58 ± 0.03	1.03 ± 0.16	**0.69 ± 0.16**	**0.25 ± 0.04**
MD-medianmean	Feasible return	0.56 ± 0.03	**0.11 ± 0.06**	0.35 ± 0.04	**0.47 ± 0.07**	0.49 ± 0.04	0.57 ± 0.06	0.52 ± 0.03	0.49 ± 0.04	**0.49 ± 0.12**
	Episode cost	1.64 ± 0.07	**0.25 ± 0.09**	1.70 ± 0.31	**0.27 ± 0.03**	1.90 ± 0.09	1.56 ± 0.11	1.43 ± 0.23	1.68 ± 0.11	**0.82 ± 0.14**
MD-hardsparse	Feasible return	0.26 ± 0.01	**0.27 ± 0.08**	0.29 ± 0.01	**0.40 ± 0.01**	0.25 ± 0.03	0.28 ± 0.01	0.28 ± 0.02	0.27 ± 0.03	**0.38 ± 0.02**
	Episode cost	1.79 ± 0.07	**0.44 ± 0.11**	2.31 ± 0.12	**0.03 ± 0.00**	1.91 ± 0.09	1.84 ± 0.09	1.95 ± 0.04	1.85 ± 0.12	**0.13 ± 0.03**

Table 1: Safety–Performance comparison on **DSRL dataset**. ($|\mathcal{D}_S| = |\mathcal{D}_P| = 10$). Results are averaged over five seeds with standard errors. Feasible return is normalized between 0 (min return) and 1 (max return); cost is normalized between 0 and 1 (cost upper bound). **Bold** marks methods satisfying constraints (mean cost $\leq$ cost upper bound). Among safety-feasible results, we highlight in **blue** those whose error bars statistically overlap with the highest mean feasible return.

Concretely, we train four policies covering distinct performance–safety combinations: `safe-high`, `safe-medium`, `unsafe-high`, and `unsafe-medium`. Cost-constrained policies (`safe-high`, `safe-medium`) are trained with PPO-Lag (Ray et al., 2019), while unconstrained policies (`unsafe-high`, `unsafe-medium`) are trained with PPO (Schulman et al., 2017). Demonstrations are then organized into datasets: $\mathcal{D}_S$ from a mixture of `safe-high` and `safe-medium`, $\mathcal{D}_P$ from a mixture of `safe-high` and `unsafe-high`, and $\mathcal{D}_U$ from all four types. Following the tabular experiments, we define **performance**($\mathcal{D}_S$) and **safeness**($\mathcal{D}_P$) as the proportions of `safe-high` demonstrations in $\mathcal{D}_S$ and $\mathcal{D}_P$, respectively.

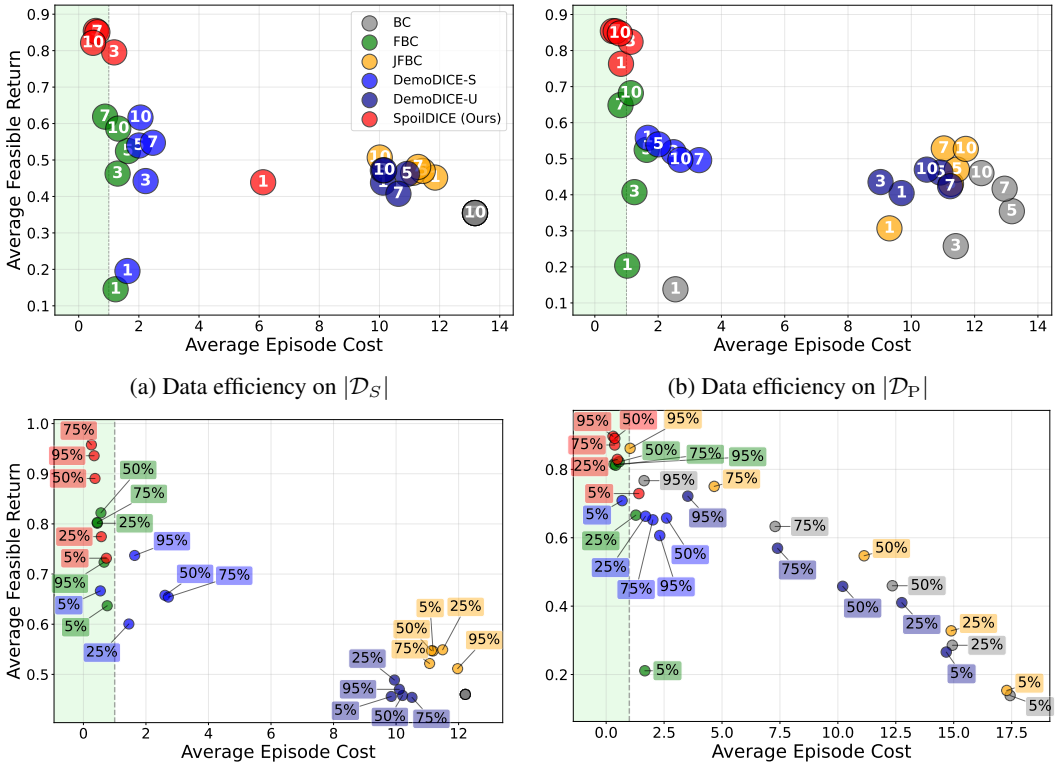
We vary the number of demonstrations of each type $k \in \{1, 3, 5, 7, 10\}$, yielding $|\mathcal{D}_P| = 2k$ or $|\mathcal{D}_S| = 2k$ depending on which set is varied, while keeping the other set and $\mathcal{D}_U$ fixed (see Section G.4.1 for experiments jointly varying $|\mathcal{D}_S| = |\mathcal{D}_P|$). Policies are then trained on these datasets for the six tasks: PointCircle1, PointCircle2, HopperVelocity, HalfCheetahVelocity, Walker2dVelocity and AntVelocity, with averaged results shown in Figures 4a and 4b.

Overall, SpoilDICE achieves the most robust safety–performance trade-off across dataset sizes. While baselines often fail to recover policies that are simultaneously safe and performant, our approach consistently maintains high feasible returns while satisfying safety constraints. SpoilDICE only struggles when $\mathcal{D}_S$ is extremely scarce, underscoring the importance of having sufficient safe demonstrations to reliably estimate the support of d_S.

5.2.3 Robustness to Data Quality

To address **Q3**, we again use the controlled datasets, varying the quality of $\mathcal{D}_P$ and $\mathcal{D}_S$ by adjusting their mixing ratios while keeping the total number of trajectories fixed. Policies are trained on these datasets for the same six tasks as in the previous section, with results shown in Figures 4c and 4d.

Baseline methods degrade sharply as demonstration quality decreases. In contrast, SpoilDICE sustains high feasible return with low cost across a wide range of quality levels, showing robustness



(a) Data efficiency on $|\mathcal{D}_S|$

(b) Data efficiency on $|\mathcal{D}_P|$

(c) Data quality dependency on **performance**$(\mathcal{D}_S)$

(d) Data quality dependency on **safeness**$(\mathcal{D}_P)$

Figure 4: Data efficiency (a,b) and sample quality (c,d) comparison across methods. Results are averaged over 6 tasks. Each subplot examines: (a) varying $|\mathcal{D}_S| = 2k$ with $k \in \{1, 3, 5, 7, 10\}$ while maintaining $|\mathcal{D}_P| = 10$; (b) scaling $|\mathcal{D}_P| = 2k$ in as the same scale while keeping $|\mathcal{D}_P| = 10$ fixed; (c) varying **performance**$(\mathcal{D}_S) \in \{5\%, 25\%, 50\%, 75\%, 95\%\}$ while fixing **safeness**$(\mathcal{D}_P) = 50\%$ and $|\mathcal{D}_P| = |\mathcal{D}_S| = 20$. (d) varying **safeness**$(\mathcal{D}_P) \in \{5\%, 25\%, 50\%, 75\%, 95\%\}$ with fixed **performance**$(\mathcal{D}_S) = 50\%$ and $|\mathcal{D}_P| = |\mathcal{D}_S| = 20$; The green shaded region denotes the feasible area satisfying cost constraints. Per-task detailed analysis is provided in Appendix G.4 and G.5.

to both performance degradation in $\mathcal{D}_S$ and safety degradation in $\mathcal{D}_P$. Performance drops are observed only in extreme cases, such as when $\mathcal{D}_P$ is dominated by unsafe trajectories (5%) or $\mathcal{D}_S$ by low-performance demonstrations (below 25%). We attribute these failures to violations of the feasibility assumption (Eq. 14), which arise when the support of d_S does not adequately cover that of safe, high-performing policies. Overall, these results highlight that SpoilDICE effectively disentangles safety and performance objectives, leveraging even partially degraded demonstrations to recover policies that remain both safe and performant, which is consistent with the observation from the tabular experiment.

Additional analyses on (a) robustness to noisy safe demonstrations and (b) sensitivity to the hyperparameter κ are provided in Section G.6, G.7, respectively.

6 Conclusion

In this work, we proposed SpoilDICE, a practical and novel framework for learning safe yet performant policies from safety-performance dual demonstrations in the offline regime, without relying on explicit reward or cost signals. We augmented the standard DICE objective with support constraints and formulated a constrained optimization problem. Through extensive experiments, we showed that SpoilDICE outperformed prior methods not only in realistic data collection scenarios but also in high-dimensional control tasks. These results indicate that SpoilDICE is an effective and robust approach, achieving strong data efficiency and maintaining performance across diverse demonstration qualities. An interesting direction for future work is to extend our framework to handle multiple safety constraints or more general multi-objective settings, thereby broadening the scope of safe offline imitation learning in complex real-world applications.

ETHICS STATEMENT

While this research aims to advance safe imitation learning, we acknowledge its potential risk of misuse. Our methods could, in principle, be applied to circumvent safety requirements or to develop policies that prioritize performance over safety. Such misuse would compromise the safeguards that the framework is designed to ensure. Moreover, in safety-critical domains such as autonomous driving, robotics, and healthcare, the deployment of imitation learning methods without rigorous validation may result in serious harm. Beyond individual applications, these risks extend to broader societal implications, as unsafe deployments could weaken public trust in AI systems. The ethical responsibility for ensuring proper use therefore lies with end users and the organizations that deploy these methods.

REPRODUCIBILITY STATEMENT

For reproducibility of our research, we provide anonymous links for experiment code and datasets:

- Code: `https://anonymous.4open.science/r/iclr2026_spoildice-214C`
- Dataset 1: `https://doi.org/10.5281/zenodo.17192925`
- Dataset 2: `https://doi.org/10.5281/zenodo.17196590`

Dataset 2 contains only SafetyHumanoidVelocity-v1, while Dataset 1 includes all other datasets.

REFERENCES

Eitan Altman. *Constrained Markov decision processes*. Routledge, 2021.

Mayank Bansal, Alex Krizhevsky, and Abhijit S. Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. In *Robotics: Science and Systems XV*, 2019.

Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Shengfan Cao, Eunhyek Joa, and Francesco Borrelli. A simple approach to constraint-aware imitation learning with application to autonomous racing. *arXiv preprint arXiv:2503.07737*, 2025.

Yusuf Umut Ciftci, Darren Chiu, Zeyuan Feng, Gaurav S Sukhatme, and Somil Bansal. SAFE-GIL: Safety guided imitation learning for robotic systems. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3559–3566, 2025.

Robert Dadashi, Leonard Hussenot, Matthieu Geist, and Olivier Pietquin. Primal wasserstein imitation learning. In *International Conference on Learning Representations*, 2020.

Bin Fang, Shidong Jia, Di Guo, Muhua Xu, Shuhuan Wen, and Fuchun Sun. Survey of imitation learning for robotic manipulation. *International Journal of Intelligent Robotics and Applications*, 3(4):362–369, 2019.

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Jeffrey Hawke, Richard Shen, Corina Gurau, Siddharth Sharma, Daniele Reda, Nikolay Nikolov, Przemysław Mazur, Sean Micklethwaite, Nicolas Griffiths, Amar Shah, et al. Urban driving with conditional imitation learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 251–257, 2020.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

Huy Hoang, Tien Mai, and Pradeep Varakantham. Sprinql: Sub-optimal demonstrations driven offline imitation learning. *Advances in Neural Information Processing Systems*, 37:136837–136872, 2024.

Adriana Hugessen, Harley Wiltzer, and Glen Berseth. Simplifying constraint inference with inverse reinforcement learning. *Advances in Neural Information Processing Systems*, 37:44501–44525, 2024.

Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

Youngsoo Jang, Geon-Hyeong Kim, Jongmin Lee, Sungryull Sohn, Byoungjip Kim, Honglak Lee, and Moontae Lee. SafeDICE: Offline safe imitation learning with non-preferred demonstrations. *Advances in Neural Information Processing Systems*, 36:74921–74951, 2023.

Jiaming Ji, Borong Zhang, Jiayi Zhou, Xuehai Pan, Weidong Huang, Ruiyang Sun, Yiran Geng, Yifan Zhong, Josef Dai, and Yaodong Yang. Safety Gymnasium: A unified safe reinforcement learning benchmark. *Advances in Neural Information Processing Systems*, 36:18964–18993, 2023.

Jiaming Ji, Jiayi Zhou, Borong Zhang, Juntao Dai, Xuehai Pan, Ruiyang Sun, Weidong Huang, Yiran Geng, Mickel Liu, and Yaodong Yang. Omnisafe: An infrastructure for accelerating safe reinforcement learning research. *Journal of Machine Learning Research*, 25(285):1–6, 2024.

Edward Johns. Coarse-to-fine imitation learning: Robot manipulation from a single demonstration. In *2021 IEEE international conference on robotics and automation (ICRA)*, pp. 4613–4619, 2021.

Geon-Hyeong Kim, Jongmin Lee, Youngsoo Jang, Hongseok Yang, and Kee-Eung Kim. LobsDICE: Offline learning from observation via stationary distribution correction estimation. *Advances in Neural Information Processing Systems*, 35:8252–8264, 2022a.

Geon-Hyeong Kim, Seokin Seo, Jongmin Lee, Wonseok Jeon, HyeongJoo Hwang, Hongseok Yang, and Kee-Eung Kim. DemoDICE: Offline imitation learning with supplementary imperfect demonstrations. In *International Conference on Learning Representations*, 2022b.

Kee-Eung Kim and Hyun Soo Park. Imitation learning via kernel mean embedding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Konwoo Kim, Gokul Swamy, Zuxin Liu, Ding Zhao, Sanjiban Choudhury, and Steven Z Wu. Learning shared safety constraints from multi-task demonstrations. *Advances in Neural Information Processing Systems*, 36:5808–5826, 2023.

Matthieu Komorowski, Leo A Celi, Omar Badawi, Anthony C Gordon, and A Aldo Faisal. The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. *Nature Medicine*, 24(11):1716–1720, 2018.

Hyeokjin Kwon, Gunmin Lee, Junseo Lee, and Songhwai Oh. Safe CoR: A dual-expert approach to integrating imitation learning and safe reinforcement learning using constraint rewards. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2893–2898, 2024.

Luc Le Mero, Dewei Yi, Mehrdad Dianati, and Alexandros Mouzakitis. A survey on imitation learning techniques for end-to-end autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 23(9):14128–14147, 2022.

Jongmin Lee, Wonseok Jeon, Byungjun Lee, Joelle Pineau, and Kee-Eung Kim. OptiDICE: Offline policy optimization via stationary distribution correction estimation. In *International Conference on Machine Learning*, pp. 6120–6130. PMLR, 2021.

Jongmin Lee, Cosmin Paduraru, Daniel J Mankowitz, Nicolas Heess, Doina Precup, Kee-Eung Kim, and Arthur Guez. COptiDICE: Offline constrained reinforcement learning via stationary distribution correction estimation. In *International Conference on Learning Representations*, 2022.

Guiliang Liu, Yudong Luo, Ashish Gaurav, Kasra Rezaee, and Pascal Poupart. Benchmarking constraint inference in inverse reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Guiliang Liu, Sheng Xu, Shicheng Liu, Ashish Gaurav, Sriram Ganapathi Subramanian, and Pascal Poupart. A comprehensive survey on inverse constrained reinforcement learning: Definitions, progress and challenges. *Transactions on Machine Learning Research*, 2025.

Zuxin Liu, Zijian Guo, Haohong Lin, Yihang Yao, Jiacheng Zhu, Zhepeng Cen, Hanjiang Hu, Wenhao Yu, Tingnan Zhang, Jie Tan, et al. Datasets and benchmarks for offline safe reinforcement learning. *Journal of Data-centric Machine Learning Research*, 2024.

Yecheng Ma, Andrew Shen, Dinesh Jayaraman, and Osbert Bastani. Versatile offline imitation from observations and examples via regularized state-occupancy matching. In *International Conference on Machine Learning*, pp. 14639–14663. PMLR, 2022.

Shehryar Malik, Usman Anwar, Alireza Aghasi, and Ali Ahmed. Inverse constrained reinforcement learning. In *International conference on machine learning*, pp. 7390–7399. PMLR, 2021.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning (CoRL)*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1678–1690. PMLR, 2021.

Shamim Nemati, Mohammad M Ghassemi, and Gari D Clifford. Optimal medication dosing from suboptimal clinical examples: A deep reinforcement learning approach. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 2978–2981, 2016. doi: 10.1109/EMBC.2016.7591355.

Guorui Quan, Guiliang Liu, et al. Learning constraints from offline demonstrations via superior distribution correction estimation. In *Forty-first International Conference on Machine Learning*, 2024.

Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*, 7(1):2, 2019.

Stephane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*. PMLR, 2010.

Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Clayton Scott and Gilles Blanchard. Novelty detection: Unlabeled data definitely help. In *Artificial Intelligence and Statistics*, pp. 464–471. PMLR, 2009.

Seokin Seo, Byung-Jun Lee, Jongmin Lee, HyeongJoo Hwang, Hongseok Yang, and Kee-Eung Kim. Mitigating covariate shift in behavioral cloning via robust stationary distribution correction. *Advances in Neural Information Processing Systems*, 37:109177–109201, 2024.

Qian Shao, Pradeep Varakantham, and Shih-Fen Cheng. Imitating cost-constrained behaviors in reinforcement learning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 34, pp. 514–522, 2024.

Ingo Steinwart, Don Hush, and Clint Scovel. A classification framework for anomaly detection. *Journal of Machine Learning Research*, 6(2), 2005.

Sriram Ganapathi Subramanian, Guiliang Liu, Mohammed Elmahgiubi, Kasra Rezaee, and Pascal Poupart. Confidence aware inverse constrained reinforcement learning. In *International Conference on Machine Learning*, pp. 14491–14512. PMLR, 2024.

Fan Xie, Alexander Chowdhury, M De Paolis Kaluza, Linfeng Zhao, Lawson Wong, and Rose Yu. Deep imitation learning for bimanual robotic manipulation. *Advances in neural information processing systems*, 33:2327–2337, 2020.

Haoran Xu, Xianyuan Zhan, Honglei Yin, and Huiling Qin. Discriminator-weighted offline imitation learning from suboptimal demonstrations. In *International Conference on Machine Learning*, pp. 24725–24742. PMLR, 2022.

He Yin, Peter Seiler, Ming Jin, and Murat Arcak. Imitation learning with stability and safety guarantees. *IEEE Control Systems Letters*, 6:409–414, 2021.

Lantao Yu, Tianhe Yu, Jiaming Song, Willie Neiswanger, and Stefano Ermon. Offline imitation learning with suboptimal demonstrations via relaxed distribution matching. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11016–11024, 2023.

Maryam Zare, Parham M Kebria, Abbas Khosravi, and Saeid Nahavandi. A survey of imitation learning: Algorithms, recent developments, and challenges. *IEEE Transactions on Cybernetics*, 2024.

Xin Zhang, Yanhua Li, Ziming Zhang, and Zhi-Li Zhang. f-GAIL: Learning f-divergence for generative adversarial imitation learning. *Advances in Neural Information Processing Systems*, 33:12805–12815, 2020.

Appendix

CONTENTS

A Problem Definition

Given a cost upper bound $\delta_c \in \mathbb{R}_{\geq 0}$ and a performance lower bound $\delta_r \in \mathbb{R}$, we define Π_{S} and Π_{P} to be the following sets of policies:

$$\Pi_{\mathrm{S},\delta_c} := \{\pi_{\mathrm{S}} \in \Pi \mid J(\pi_{\mathrm{S}}; c) \leq \delta_c\} \quad \text{and} \quad \Pi_{\mathrm{P},\delta_r} := \{\pi_{\mathrm{P}} \in \Pi \mid J(\pi_{\mathrm{P}}; r) \geq \delta_r\}.$$

Intuitively, Π_{S} is the set of safe policies, and Π_{P} that of performant policies. Also, for a distribution d on state-action pairs, we write $\mathrm{supp}(d)$ for the support of d, i.e., $\mathrm{supp}(d) = \{(s,a) \mid d(s,a) > 0\}$.

Definition 1 (IL from Safety–Performance Dual Demonstrations (IL-SP))**.** *Let $\delta_c \in \mathbb{R}_{\geq 0}$ and $\delta_r \in \mathbb{R}$ denote a cost upper bound and a performance lower bound, respectively. Let $\pi_{\mathrm{S}} \in \Pi_{\mathrm{S},\delta_c}$ and $\pi_{\mathrm{P}} \in \Pi_{\mathrm{P},\delta_r}$ denote a safe policy and a performant policy, with stationary distributions d_{S} and d_{P}. We assume the feasibility condition*

$$\overline{\Pi} := \{\pi \in \Pi \mid \mathrm{supp}(d_\pi) \subseteq \mathrm{supp}(d_{\mathrm{S}}) \cap \mathrm{supp}(d_{\mathrm{P}})\} \neq \emptyset, \tag{14}$$

where d_π is the stationary distribution of policy π. Given finite demonstration datasets

$$\mathcal{D}_{\mathrm{S}} = \{\tau_{\mathrm{S}}^{(1)}, \ldots, \tau_{\mathrm{S}}^{(m)}\}, \quad \tau_{\mathrm{S}}^{(i)} \sim \pi_{\mathrm{S}}, \qquad \mathcal{D}_{\mathrm{P}} = \{\tau_{\mathrm{P}}^{(1)}, \ldots, \tau_{\mathrm{P}}^{(n)}\}, \quad \tau_{\mathrm{P}}^{(i)} \sim \pi_{\mathrm{P}},$$

where each trajectory $\tau = \{s_0, a_0, s_1, a_1, \ldots\}$ is generated by $s_0 \sim \rho_0$, $a_t \sim \pi(s_t)$, and $s_{t+1} \sim T(s_t, a_t)$, the IL-SP problem *seeks a policy π that minimizes*

$$\min_{\pi \in \Pi} \mathbb{D}(d_\pi \| d_{\mathrm{P}}) \quad \textit{subject to} \quad \mathrm{supp}(d_\pi) \subseteq \mathrm{supp}(d_{\mathrm{S}}). \tag{15}$$

The non-emptiness of $\overline{\Pi}$ ensures that some π satisfies the support constraint in the problem and also has a finite value when $\mathbb{D}$ is the KL divergence; these imply the feasibility of the above optimization problem in this case. If interaction with the environment is allowed during optimization, the problem is referred to as the online setting; otherwise, it is the offline setting.

Note that IL-SP problem is closely related to cost-constrained IL (Shao et al., 2024). In particular, a solution to IL-SP corresponds to a feasible solution of the cost-constrained IL problem with a hard constraint, where the cost function is non-negative ($c \geq 0$) and the upper bound is fixed to $\delta_c = 0$:

$$\min_{\pi} \quad \mathbb{D}(d_\pi \| d_{\mathrm{P}}) \quad \text{subject to} \quad J(\pi; c) = 0.$$

This is because the safety condition $J(\pi; c) = 0$ follows from our support constraint $\mathrm{supp}(d_\pi) \subseteq \mathrm{supp}(d_{\mathrm{S}})$. To see why, note that $J(\pi_{\mathrm{S}}; c) \leq \delta_c = 0$. Together with the non-negativity of c, this implies that $c(s,a) = 0$ for all $(s,a) \in \mathrm{supp}(d_{\mathrm{S}})$. Thus, also by our support constraint, we have $c(s,a) = 0$ for all $(s,a) \in \mathrm{supp}(d_\pi)$, which gives $J(\pi; c) = \mathbb{E}_{d_\pi}[c(s,a)] = 0$, that is, π satisfies the hard constraint.

In practice, demonstrations may be collected from multiple safe policies and multiple performant policies, instead of just one π_{S} and one π_{P}. This general case is also covered by our setup. A finite set of demonstrations from multiple policies can be viewed as a demonstration set of a single policy obtained by a convex combination of the original multiple policies. Furthermore, if all those original policies are safe (i.e., they are all in $\Pi_{\mathrm{S},\delta_c}$), so is their convex combination. Similarly, if the policies are performant (i.e., they belong to $\Pi_{\mathrm{P},\delta_r}$), so is their convex combination. Thus, the general case can be understood as an instance of our problem where π_{S} and π_{P} are appropriate convex combinations of multiple safe and performant policies, respectively.

IL-SP problem with a trivial safe-support When d_S has full support over the state–action space, i.e., $d_S(s, a) > 0$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$, the support constraint in Eq. 15 becomes vacuous:

$$\text{supp}(d_\pi) \subseteq \text{supp}(d_S) = \mathcal{S} \times \mathcal{A},$$

and therefore it imposes no restriction on the feasible set of stationary distributions d_π. This situation is directly analogous to constrained RL settings in which the set of cost-feasible policies coincides with the entire policy class, thereby reducing the constrained problem to its unconstrained counterpart. Importantly, this does not invalidate IL-SP objective; rather, it reflects that the provided *safe* demonstrations fail to encode any meaningful safety signal. In this case, IL-SP naturally collapses to unconstrained stationary-distribution matching (equivalent to DemoDICE without behavior regularization), i.e.,

$$\min_{d \in \mathcal{Q}} \mathbb{D}(d \,\|\, d_P)$$

where $\mathcal{Q}$ is a set of stationary distributions. This observation underscores the practical importance of collecting informative safe demonstrations, much like the necessity of specifying non-trivial cost constraints in constrained RL.

IL-SP with no support overlap If the safe and performant stationary distributions, d_S and d_P, have

$$\text{supp}(d_S) \cap \text{supp}(d_P) = \emptyset,$$

then the feasible region of IL–SP becomes severely restricted. Under the IL–SP formulation

$$\min_{d \in \mathcal{Q}} \mathbb{D}(d \,\|\, d_P) \quad \text{s.t.} \quad \text{supp}(d) \subseteq \text{supp}(d_S),$$

any feasible d must satisfy $d(s, a) = 0$ outside $\text{supp}(d_S)$. Since d_P assigns mass exclusively outside this region, the problem effectively becomes

$$\min_{d \in \mathcal{Q}:\, \text{supp}(d) \subseteq \text{supp}(d_S)} \mathbb{D}(d \,\|\, d_P),$$

which simply selects the feasible d that yields the smallest divergence to d_P. Since d_P places mass entirely outside this region, the optimization reduces to select the *closest* one to d_P under $\mathbb{D}$ among all stationary distributions supported on $\text{supp}(d_S)$. Thus, IL-SP will chooses the closest stationary distribution inside the safe support, even when d_P lies entirely outside that region. This limitation is also inherent to constrained optimization and is directly analogous to constrained RL. When the constraint is overly strict, no policy that attains sufficiently high return can satisfy the constrained RL objective.

B Detailed Discussions on Related Work

DICE Method	Target Offline Problem	Reward	Cost	$\mathcal{D}_\mathrm{P}$	$\mathcal{D}_\mathrm{S}$	$\mathcal{D}_\mathrm{N}$	Safety	Constraints
OptiDICE (Lee et al., 2022)	RL	✓	–	–	–	–	–	Distribution
DemoDICE (Kim et al., 2022b), SMODICE (Ma et al., 2022), RelaxDICE (Yu et al., 2023)	IL	–	–	✓	–	–	–	Distribution
SafeDICE (Jang et al., 2023)	IL	–	–	–	–	✓	✓	Distribution
COptiDICE (Lee et al., 2022)	Constrained RL	✓	✓	–	–	–	✓	Distribution, Cost (true)
ICSDICE (Quan et al., 2024)	Inverse Constrained RL	✓	–	–	✓	–	✓	Distribution, Cost (inferred)
SpoilDICE (Ours)	IL–SP	–	–	✓	✓	–	✓	Support

Table A: Structural comparison of DICE methods for offline problems in terms of required learning signals, safety constraints, and dataset requirements. $\mathcal{D}_\mathrm{N}$ indicates a set of non-preferred demonstrations, which are corresponding to unsafe demonstrations in our performance–safety perspective.

In this section, we provide a detailed discussion of DICE-based approaches developed for offline learning problems that are closely related to our setting. Table A summarizes representative methods in terms of required learning signals, safety considerations, and dataset requirements.

DICE methods have been widely studied in both RL and IL because they bypass explicit sequential modeling of MDP dynamics by optimizing over stationary distributions constrained to satisfy the Bellman flow condition. This perspective leads to linear programming objectives with tractable convex dual forms, making DICE formulations particularly appealing for offline settings.

A representative offline RL variant is OptiDICE (Lee et al., 2022), whose objective is

$$\max_{d \in \mathcal{Q}} \ \mathbb{E}_d[r(s,a)] \ - \ \alpha \, D_f(d \,\|\, d_\mathrm{U}),$$

where $\mathcal{Q}$ denotes the set of stationary distributions. The f-divergence regularizer plays a central role by enforcing conservatism, preventing the learned distribution from drifting outside the support of the offline dataset, which is a standard requirement for stable offline RL. DemoDICE (Kim et al., 2022b) adapts this idea to offline IL using the objective

$$\max_{d \in \mathcal{Q}} \ - D_\mathrm{KL}(d \,\|\, d_\mathrm{P}) \ - \ \alpha \, D_\mathrm{KL}(d \,\|\, d_\mathrm{U}),$$

which encourages the learned stationary distribution to remain close to both the performant demonstrations d_P and the offline dataset d_U. As shown in Table A, DemoDICE also rely on distributional proximity to d_U, which may be problematic in IL–SP, if the offline dataset contains suboptimal or unsafe behavior, forcing closeness to d_U can degrade either safety or performance.

As discussed in Section 2, replacing d_U with d_S (safe demonstrations) in DemoDICE can partially mitigate safety concerns, but still requires staying close to possibly low-quality safe demonstrations, making the method sensitive to data quality.

To address this limitation, SpoilDICE explicitly replaces the distributional regularizer with a support constraint:

$$\max_{d \in \mathcal{Q}} \ -D_\mathrm{KL}(d \,\|\, d_\mathrm{P}) \quad \text{subject to} \quad \mathrm{supp}(d) \subseteq \mathrm{supp}(d_\mathrm{S}).$$

This formulation seeks the stationary distribution that is closest to d_P while remaining entirely within the safe support defined by d_S. As summarized in Table A, SpoilDICE is the only DICE-based approach that enforces safety purely through a support constraint without requiring reward signals, cost signals, or non-preferred demonstrations. By avoiding distributional penalties toward d_S, SpoilDICE becomes significantly less sensitive to the quality of safe demonstrations, a property reflected in our empirical analyses.

C PROOFS OF PROPOSITIONS

Rewrite an objective (12):

$$L(w, \nu, \lambda) = (1-\gamma)\mathbb{E}_{s\sim\rho_0}[\nu(s)] + \mathbb{E}_{(s,a)\sim d_U}\left[w(s,a)(A_{\nu,\lambda}(s,a) - \log w(s,a))\right] - \lambda.$$

Proposition 1. *(Strong Duality) $L(w, \nu, \lambda)$ satisfies strong duality which ensures:*

$$\max_{w\geq 0}\min_{\nu,\lambda} L(w,\nu,\lambda) = \min_{\nu,\lambda}\max_{w\geq 0} L(w,\nu,\lambda)$$

Proof. Without loss of generality, we can assume that every state $s \in \mathcal{S}$ is reachable. In such MDPs, there always exists a strictly feasible d, thus Slater's condition is satisfied, implying that the strong duality holds (Boyd & Vandenberghe, 2004). □

Proposition 2 (Closed-form solution)**.** *For any dual variables ν and λ, the inner maximization*

$$\max_{w\geq 0} L(w,\nu,\lambda)$$

admits the closed-form solution

$$w^*_{\nu,\lambda}(s,a) = \exp\big(A_{\nu,\lambda}(s,a) - 1\big).$$

Proof. Recall the following lemma (Proposition 1 in (Lee et al., 2022)):

Lemma. *Let f be a convex function (e.g. for KL divergence, $f(x) = x\log x$) with strictly increasing derivative f'. Given ν, λ, and $\alpha \geq 0$, define*

$$e_{\nu,\lambda}(s,a) := r(s,a) - \lambda c(s,a) + \gamma\,\mathbb{E}_{s'\sim T(s,a)}[\nu(s')] - \nu(s).$$

Then, the maximization

$$\max_{w\geq 0}\ \mathbb{E}_{(s,a)\sim d_U}\big[\, w(s,a)\, e_{\nu,\lambda}(s,a) - \alpha\, f(w(s,a))\big]$$

is achieved by

$$w^*_{\nu,\lambda}(s,a) = \big(f'\big)^{-1}\big(\tfrac{1}{\alpha}\, e_{\nu,\lambda}(s,a)\big)_+,$$

where $(\cdot)_+ = \max(0,\cdot)$.

Applying this lemma in our setting: we substitute

$$r = \log r_P, \quad c = -\mathbb{1}_{d_S}, \quad \alpha = 1, \quad \delta_c = -1,$$

which yields $e_{\nu,\lambda} = A_{\nu,\lambda}$ and $L = \mathcal{L}$ in the canonical form. Since for KL divergence one has $f(x) = x\log x$ and thus $(f')^{-1}(y) = \exp(y-1)$, we conclude

$$w^*_{\nu,\lambda}(s,a) = \exp\big(A_{\nu,\lambda}(s,a) - 1\big).$$

□

D Details on Practical Algorithm

D.1 Discriminator-based Estimation

Following Kim et al. (2022b), we train discriminators $\hat{C}_{\mathrm{P}}, \hat{C}_S : \mathcal{S} \times \mathcal{A} \to [0,1]$ with neural networks to estimate the density ratio $r_{\mathrm{P}} = d_{\mathrm{P}}/d_{\mathrm{U}}$ and $r_{\mathrm{S}} = d_{\mathrm{S}}/d_{\mathrm{U}}$ via the following objectives:

$$\hat{C}_{\mathrm{P}} \in \underset{C:\mathcal{S}\times\mathcal{A}\to[0,1]}{\arg\max} \mathbb{E}_{d_{\mathrm{P}}}[\log C(s,a)] + \mathbb{E}_{d_{\mathrm{U}}}[\log(1 - C(s,a))] \tag{16}$$

$$\hat{C}_{\mathrm{S}} \in \underset{C:\mathcal{S}\times\mathcal{A}\to[0,1]}{\arg\max} \mathbb{E}_{d_{\mathrm{S}}}[\log C(s,a)] + \mathbb{E}_{d_{\mathrm{U}}}[\log(1 - C(s,a))] \tag{17}$$

The solutions correspond to $C^*_{\mathrm{P}}(s,a) = \frac{d_{\mathrm{P}}(s,a)}{d_{\mathrm{P}}(s,a)+d_{\mathrm{U}}(s,a)}, C^*_{\mathrm{S}}(s,a) = \frac{d_{\mathrm{S}}(s,a)}{d_{\mathrm{S}}(s,a)+d_{\mathrm{U}}(s,a)}$ respectively. From $\hat{C}_{\mathrm{P}}, \hat{C}_{\mathrm{S}}$, we estimate the density ratios $\hat{r}_{\mathrm{P}}, \hat{r}_{\mathrm{S}}$ as follows:

$$\hat{r}_{\mathrm{P}}(s,a) = \frac{d_{\mathrm{P}}(s,a)}{d_{\mathrm{U}}(s,a)} = \frac{\hat{C}_{\mathrm{P}}(s,a)}{1 - \hat{C}_{\mathrm{P}}(s,a)}, \quad \hat{r}_{\mathrm{S}}(s,a) = \frac{d_{\mathrm{S}}(s,a)}{d_{\mathrm{U}}(s,a)} = \frac{\hat{C}_{\mathrm{S}}(s,a)}{1 - \hat{C}_{\mathrm{S}}(s,a)} \tag{18}$$

D.2 Numerical Stability

To improve numerical stability during optimization arised from a summation of exponentials, we adopt an alternative log-sum-exp, as proposed by Kim et al. (2022b).

Proposition 3. *(Kim et al. (2022b)) Define an alternative objective $\tilde{L}(\nu, \lambda)$ as follows:*

$$\tilde{L}(\nu,\lambda) := (1-\gamma)\mathbb{E}_{p_0}[\nu(s)] + \log \mathbb{E}_{(s,a)\sim d_D}\left[\exp\left(A_{\nu,\lambda}(s,a)\right)\right] - \lambda \tag{19}$$

Then, the following holds:

$$\min_{\nu} L(\nu,\lambda) = \min_{\nu} \tilde{L}(\nu,\lambda).$$

D.3 Policy Extraction

In tabular settings, we can directly recover the policy from the optimal stationary distribution $d^*(s,a) = w^*_{\nu,c}(s,a), d_{\mathrm{U}}(s,a)$ as:

$$d^*(s,a) = w^*_{\nu,\lambda}(s,a) d_{\mathrm{U}}(s,a), \quad \pi(a|s) = \frac{d^*(s,a)}{\sum_{\bar{a}\in\mathcal{A}} d^*(s,\bar{a})}.$$

However, in continuous spaces, computing $d^*(s) = \int d^*(s,a)da$ is intractable. Thus, following Lee et al. (2021); Kim et al. (2022b), we extract the policy via weighted behavioral cloning:

$$\min_{\pi} \quad J(\pi; w^*_{\nu,\lambda}) := \mathbb{E}_{(s,a)\sim d_{\mathrm{U}}}\left[-w^*_{\nu,\lambda}(s,a)\log\pi(a|s)\right]. \tag{20}$$

D.4 Algorithm Summary

We summarize a practical version of the SpoilDICE algorithm in **Algorithm 1**.

Algorithm 1: SpoilDICE (**S**afe-**p**erformant **o**ffline **i**mitation **l**earning via **DICE**)

Input: Safe, performant, unlabeled demonstrations sets $\mathcal{D}_{\mathrm{S}}, \mathcal{D}_{\mathrm{P}}, \mathcal{D}_{\mathrm{U}}$, an initial state set $\mathcal{D}_0$
discriminator networks $C_{\mathrm{P}}, C_{\mathrm{S}}$ with parameters $\zeta_{\mathrm{P}}, \zeta_{\mathrm{S}}$, percentile cutoff κ (e.g. 0.01)
neural networks $\nu_\theta(s)$ policy network π_ϕ with parameters θ, ϕ, Lagrangian λ

Initialize $\zeta_{\mathrm{P}}, \zeta_{\mathrm{S}}$.
Pretrain $\hat{C}_{\mathrm{P}}$ and $\hat{C}_{\mathrm{S}}$ by optimizing objectives (16,17)
Obtain $\hat{r}_{\mathrm{P}}, \hat{r}_{\mathrm{S}}$ from Eq. (18)
Obtain $\hat{\mathbb{1}}_{\mathrm{S}}$ with κ from Eq. (13).
Initialize θ, ϕ, ψ.
for *each iteration* **do**
  Sample mini-batches from $\mathcal{D}_{\mathrm{P}}, \mathcal{D}_{\mathrm{S}}, \mathcal{D}_{\mathrm{U}}, \mathcal{D}_0$ respectively.
  Compute θ-gradient, ϕ-gradient to optimize the obj. (19):

$$g_\theta \approx \nabla_\theta \tilde{L}(\nu_\theta, \lambda), \quad g_\lambda \approx \nabla_\lambda L(\nu_\theta, \lambda)$$

  Compute ψ-gradient to optimize the obj. (20):

$$g_\phi \approx \nabla_\phi J(\pi_\phi; w^*_{\nu_\theta, \lambda}).$$

  Perform SGD updates:

$$\theta \leftarrow \theta - \eta g_\theta, \quad \phi \leftarrow \phi - \eta g_\phi, \quad \lambda \leftarrow \lambda - \eta g_\lambda.$$

end
Output: $\nu_\theta \approx \nu^*, \pi_\phi \approx \pi^*$

E Details of Tabular Experiments

E.1 CMDP Configurations

We configure the constrained grid-world as an MDP defined by:

- $\mathcal{S}$: agent locations ($|\mathcal{S}| = 122 = 11 \times 11 + 1$ with an absorbing state).
- $\mathcal{A}$: agent actions, $\{\uparrow, \downarrow, \leftarrow, \rightarrow\}$.
- $T : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$: with probability 0.99, the agent moves in the chosen direction, and with probability 0.01 transitions to one of the other directions.
- r: reward of 1 when the agent reaches the goal.
- c: cost of 1 when the agent enters the constraint zone which differs for each seed.
- $\gamma = 0.95$.

We set the cost upper bound to $\delta_c = 10^{-4}$. To obtain demonstration distributions, we define:

- $d^*_{\mathcal{M}}$: stationary distribution of the unconstrained MDP-optimal policy.
- $d^*_{\mathcal{M}_c}$: stationary distribution of the CMDP-optimal policy.
- $d^o_{\mathcal{M}_c}$: stationary distribution of a cost-satisfying policy with performance level $o \in [0, 1]$, where $o = 0$ corresponds to a random policy and $o = 1$ to a high-performing policy.

For comparison, we construct d_{P} as a convex combination of $d^*_{\mathcal{M}}$ and $d^*_{\mathcal{M}_c}$:

$$d_{\mathrm{P}} = (1 - \xi)\, d^*_{\mathcal{M}} + \xi\, d^*_{\mathcal{M}_c}, \quad \xi \in [0, 1],$$

where ξ controls the safeness of d_{P}. For d_{S}, we use $d^o_{\mathcal{M}_c}$ with performance level o.

Finally, we obtain d according to the objectives of BC, Filtered BC, DemoDICE, and SpoilDICE, described in Table B.

Corresponding Method	**Distribution-level Original Objective**
BC	$d = d_{\mathrm{P}}$
Filtered BC (FBC)	$d(s,a) = \frac{d_{\mathrm{P}}(s,a)\mathbb{1}_{d_{\mathrm{S}}}(s,a)}{Z} \quad \forall (s,a)$ where $Z = \sum_{(s,a)\in \mathrm{supp}(d_{\mathrm{S}})} d_{\mathrm{P}}(s,a)$
DemoDICE	$d \in \arg\min_d D_{\mathrm{KL}}(d \Vert d_{\mathrm{P}}) + \alpha D_{\mathrm{KL}}(d \Vert d_{\mathrm{S}})$
SpoilDICE (Ours)	$d \in \arg\min_d D_{\mathrm{KL}}(d \Vert d_{\mathrm{P}}) \quad$ s.t. $\quad \sum_{s,a} \mathbb{1}_{d_{\mathrm{S}}}(s,a) d(s,a) = 1$

Table B: Distribution-level objective comparisons of methods for tabular experiment.

E.2 Detailed Evaluation

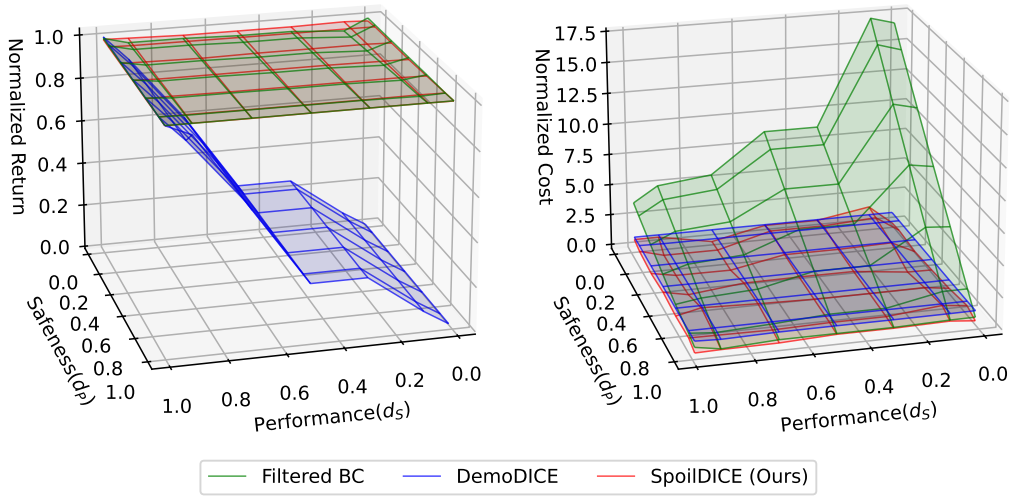


Figure A: Comparison of methods on distribution-level objectives in the constrained grid-world task. Returns are normalized between 0 (random policy) and 1 (CMDP optimal), while costs are normalized with respect to the cost threshold in $[0, 1]$. (left) Normalized return as the quality of d_{S} and d_{P} varies. (right) Normalized cost under the same conditions.

From each distribution d, we derive a policy π defined by $\pi(a|s) = d(s,a) \big/ \sum_{\bar{a}} d(s,\bar{a}),\ \forall s \in \mathcal{S}$, and evaluate π with respect to both rewards and costs. For hyperparameters, DemoDICE uses $\alpha \in [10^{-2}, 10^{2}]$, while Filtered BC and SpoilDICE use support thresholds that define $\mathbb{1}_{d_{\mathrm{S}}}$ by filtering out low-probability state–action pairs under d_{S}, with values ranging from 10^{-10} to 10^{-4}. For each problem setting, we report the highest return among feasible policies (i.e., those satisfying $J(\pi; c) \leq \delta_c$); if no feasible policy exists, we instead report the policy with the lowest expected cost. All results are averaged over 50 random seeds, with the constraint zone randomized for each seed.

Figure A presents the normalized return and cost as functions of the safeness level of $\mathcal{D}_{\mathrm{S}}$ and the performance level of $\mathcal{D}_{\mathrm{P}}$. DemoDICE exhibits a sharp degradation in return as the demonstrations become less safe or less performant, highlighting its vulnerability to conflicting regularization to safe demonstration. Filtered BC, while maintaining relatively stable returns, incurs substantially higher costs, indicating that filtering alone is insufficient to guarantee safety. In contrast, our method, SpoilDICE, consistently achieves high returns while keeping costs low across the entire spectrum of safeness–performance trade-offs. These results demonstrate that SpoilDICE effectively leverages dual demonstrations to balance performance and safety, outperforming baselines in both metrics.

F Dataset Descriptions

In Section F.1, we first describe how the original DSRL datasets (Liu et al., 2024) were reorganized and adapted to the IL-SP setting by labeling trajectories into four dataset types based on predefined return and cost thresholds, which are used in the experiments presented in Section 5.2.1. We then summarize how policies were obtained and how the data collection was performed to obtain replay datasets and controlled datasets in Safety Gymnasium benchmark (Ji et al., 2023). The detailed descriptions are in Section F.2. For brevity, we omit environment version postfixes (e.g., '-v0', '-v1') in the experiment descriptions.

We conducted experiments on 21 tasks from the DSRL benchmark and nine tasks from the Safety Gymnasium benchmark. Together, these tasks span a diverse range of evaluation settings, from low-dimensional to high-dimensional control problems, with varying safety levels.

F.1 DSRL Datasets

DSRL offline datasets (Liu et al., 2024) were generated by mixing demonstrations of policies trained with various safe reinforcement learning algorithms, yielding a large pool of pre-collected trajectories. We first defined a cost threshold (one of $\{20, 40, 80\}$ as suggested in the original DSRL paper) and classified each trajectory as either safe or unsafe based on whether its cumulative cost fell below or exceeded this threshold. We set the return threshold to the 90% quantile of returns from safe trajectories. Using cost and return threshold, the trajectories were reorganized into `performant demo`($\mathcal{D}_{\mathrm{P}}$), `safe demo`($\mathcal{D}_{\mathrm{S}}$), `non-preferred demo`($\mathcal{D}_{\mathrm{N}}$), and `unlabeled demo`($\mathcal{D}_{\mathrm{U}}$). Note that the `non-preferred demo` subset is specifically used in the SafeDICE experiments. Table C shows the descriptions of the DSRL tasks .

Task	Obs. Dim	Act. Dim	Cost Threshold	Return Threshold	Max Return	Min Return
Safety Gymnasium						
PointCircle1	28	2	40	44.85	61.73	20.07
PointGoal1	60	2	40	21.67	30.07	0.01
PointButton1	76	2	40	27.12	41.19	0.01
PointCircle2	28	2	80	43.10	54.02	20.48
PointGoal2	60	2	80	20.40	27.72	0.00
PointButton2	76	2	80	29.75	42.90	0.01
HopperVelocity	11	3	40	1607.84	1911.40	37.05
Walker2dVelocity	17	6	40	2718.04	3418.22	18.67
HalfCheetahVelocity	17	6	40	2563.55	2806.93	5.75
AntVelocity	27	8	40	2743.78	2976.28	6.15
Bullet Safety Gym						
BallRun	8	2	40	556.19	1327.45	26.34
BallCircle	7	2	40	712.45	881.46	0.38
CarRun	8	2	20	566.21	574.65	204.29
CarCircle	7	2	40	435.21	534.31	3.48
DroneRun	18	4	40	411.47	682.83	10.56
DroneCircle	17	4	40	713.46	996.39	207.79
AntRun	34	8	40	664.83	955.48	0.00
AntCircle	33	8	40	225.71	460.71	0.02
MetaDrive						
MD-easydense	259	2	40	325.86	425.82	19.71
MD-medianmean	259	2	20	262.39	269.82	17.03
MD-hardsparse	259	2	20	311.44	486.81	17.25

Table C: Descriptions of Tasks in DSRL Datasets.

We detail the data labeling process for each demonstration category below:

- `safe demo`: demonstrations whose cumulative cost remains below the predefined cost threshold, regardless of their return.
- `performant demo`: demonstrations whose return exceeds the predefined return threshold, regardless of their cost.
- `non-preferred demo`: demonstrations whose cumulative cost exceeds the cost threshold (i.e., unsafe trajectories).
- `unlabeled demo`: the full set of offline trajectories.

Tables D, E, and F summarize the statistics of the DSRL datasets across 21 tasks with three different environments, reporting the safe ratio, performant ratio, safe-performant ratio, and the total number of trajectories for each demonstration category.

Task	Dataset	Safe Ratio	Performant Ratio	Safe-Performant Ratio	# traj.
Safety Gymnasium					
PointCircle1	safe demo	1.0	0.10	0.10	214
	performant demo	0.03	1.0	0.03	793
	nonpreferred demo	0.0	0.87	0.0	884
	unlabeled demo	0.19	0.72	0.02	1098
PointGoal1	safe demo	1.0	0.10	0.10	1364
	performant demo	0.44	1.0	0.44	313
	nonpreferred demo	0.0	0.27	0.0	658
	unlabeled demo	0.67	0.15	0.07	2022
PointButton1	safe demo	1.0	0.10	0.10	783
	performant demo	0.24	1.0	0.24	327
	nonpreferred demo	0.0	0.17	0.0	1485
	unlabeled demo	0.35	0.14	0.03	2268
PointCircle2	safe demo	1.0	0.10	0.10	349
	performant demo	0.09	1.0	0.09	409
	nonpreferred demo	0.0	0.68	0.0	546
	unlabeled demo	0.39	0.46	0.04	895
PointGoal2	safe demo	1.0	0.10	0.10	1963
	performant demo	0.38	1.0	0.38	525
	nonpreferred demo	0.0	0.22	0.0	1479
	unlabeled demo	0.57	0.15	0.06	3442
PointButton2	safe demo	1.0	0.10	0.10	1513
	performant demo	0.39	1.0	0.39	386
	nonpreferred demo	0.0	0.13	0.0	1775
	unlabeled demo	0.46	0.12	0.05	3288
HopperVelocity	safe demo	1.0	0.10	0.10	461
	performant demo	0.09	1.0	0.09	501
	nonpreferred demo	0.0	0.26	0.0	1779
	unlabeled demo	0.21	0.22	0.02	2240
Walker2dVelocity	safe demo	1.0	0.10	0.10	1020
	performant demo	0.17	1.0	0.17	584
	nonpreferred demo	0.0	0.28	0.0	1709
	unlabeled demo	0.37	0.21	0.04	2729
HalfCheetahVelocity	safe demo	1.0	0.10	0.10	664
	performant demo	0.13	1.0	0.13	505
	nonpreferred demo	0.0	0.24	0.0	1831
	unlabeled demo	0.27	0.20	0.03	2495
AntVelocity	safe demo	1.0	0.10	0.10	802
	performant demo	0.10	1.0	0.10	782
	nonpreferred demo	0.0	0.48	0.0	1447
	unlabeled demo	0.36	0.35	0.04	2249

Table D: Statistics for 4 types of datasets for Safety Gymnasium environment in DSRL Datasets.

Task	Dataset	Safe Ratio	Performant Ratio	Safe-Performant Ratio	# traj.
Bullet Safety Gym					
BallRun	safe demo	1.0	0.10	0.10	315
	performant demo	0.07	1.0	0.07	488
	nonpreferred demo	0.0	0.73	0.0	625
	unlabeled demo	0.34	0.52	0.03	940
BallCircle	safe demo	1.0	0.10	0.10	385
	performant demo	0.12	1.0	0.12	338
	nonpreferred demo	0.0	0.60	0.0	501
	unlabeled demo	0.43	0.38	0.04	886
CarRun	safe demo	1.0	0.10	0.10	470
	performant demo	0.52	1.0	0.52	91
	nonpreferred demo	0.0	0.24	0.0	181
	unlabeled demo	0.72	0.14	0.07	651
CarCircle	safe demo	1.0	0.10	0.10	634
	performant demo	0.10	1.0	0.10	633
	nonpreferred demo	0.0	0.70	0.0	816
	unlabeled demo	0.44	0.44	0.04	1450
DroneRun	safe demo	1.0	0.10	0.10	933
	performant demo	0.14	1.0	0.14	682
	nonpreferred demo	0.0	0.56	0.0	1057
	unlabeled demo	0.47	0.34	0.05	1990
DroneCircle	safe demo	1.0	0.10	0.10	728
	performant demo	0.06	1.0	0.06	1219
	nonpreferred demo	0.0	0.96	0.0	1195
	unlabeled demo	0.38	0.63	0.04	1923
AntRun	safe demo	1.0	0.10	0.10	654
	performant demo	0.09	1.0	0.09	727
	nonpreferred demo	0.0	0.57	0.0	1162
	unlabeled demo	0.36	0.40	0.04	1816
AntCircle	safe demo	1.0	0.10	0.10	1611
	performant demo	0.05	1.0	0.05	3244
	nonpreferred demo	0.0	0.75	0.0	4117
	unlabeled demo	0.28	0.57	0.03	5728

Table E: Statistics for 4 types of datasets for Bullet Safety Gym environment in DSRL Datasets.

Task	Dataset	Safe Ratio	Performant Ratio	Safe-Performant Ratio	# traj.
MetaDrive					
easydense	safe demo	1.0	0.10	0.10	675
	performant demo	0.21	1.0	0.21	331
	nonpreferred demo	0.0	0.81	0.0	325
	unlabeled demo	0.68	0.33	0.07	1000
mediummean	safe demo	1.0	0.10	0.10	576
	performant demo	0.25	1.0	0.25	230
	nonpreferred demo	0.0	0.41	0.0	424
	unlabeled demo	0.58	0.23	0.06	1000
hardsparse	safe demo	1.0	0.10	0.10	523
	performant demo	0.18	1.0	0.18	287
	nonpreferred demo	0.0	0.49	0.0	477
	unlabeled demo	0.52	0.29	0.05	1000

Table F: Statistics for 4 types of datasets for MetaDrive environment in DSRL Datasets.

F.2 Replay and Controlled Datasets

We used algorithms implemented in OmniSafe (Ji et al., 2024), a framework that provides implementations of various safe reinforcement learning algorithms for training on the Safety Gymnasium benchmark (Ji et al., 2023). Among these algorithms, we adopted implementations of PPO (Schulman et al., 2017) for cost-unconstrained Markov Decision Processes (MDPs), and PPO-Lag (Ray et al., 2019) for constrained Markov Decision Processes (CMDPs) with explicit cost upper bounds.

Task	Obs. Dim	Act. Dim	Cost upper bound
Safe Navigation Tasks			
SafetyPointCircle1	28	2	25
SafetyPointCircle2	28	2	25
Safe Velocity Tasks			
SafetyHopperVelocity	11	3	25
SafetyWalker2dVelocity	17	6	25
SafetyHalfCheetahVelocity	17	6	25
SafetyAntVelocity	27	8	25
SafetyHumanoidVelocity	376	17	25
Safe Manipulation Tasks			
ShadowHandOverSafeJoint	398	40	40
ShadowHandOverSafeFinger	398	40	40

Table G: Descriptions of Tasks in Safety Gymnasium.

For the replay datasets, each task was trained using the PPO-Lag method. During training, at every epoch we rolled out trajectories using the current policy and stored a fixed amount of trajectories into a replay buffer, from which the replay datasets were constructed. In particular, Table H summarizes the statistics of the replay datasets across environments, reporting the safe ratio, performant ratio, safe-performant ratio, and the number of trajectories.

For the experiments in Table K, we subsample datasets $\mathcal{D}_{\mathrm{P}}, \mathcal{D}_{\mathrm{S}}, \mathcal{D}_{\mathrm{U}}$ from a single replay buffer with the following configurations:

- PointCircle1, PointCircle2, HopperVelocity, Walker2dVelocity, HalfCheetahVelocity, AntVelocity: $|\mathcal{D}_{\mathrm{P}}|$=5, $|\mathcal{D}_{\mathrm{S}}|$=50,$|\mathcal{D}_{\mathrm{U}}|$=500.
- HumanoidVelocity, HandOverSafeJoint, HandOverSafeFinger: $|\mathcal{D}_{\mathrm{P}}|$=100, $|\mathcal{D}_{\mathrm{P}}| = 1000$, $|\mathcal{D}_{\mathrm{U}}| = 10000$.

For the controlled datasets, we adopted PPO for cost-unconstrained Markov Decision Processes (MDPs) and PPO-Lag for constrained Markov Decision Processes (CMDPs) with explicit cost upper bounds. Using these algorithms, we trained near-optimal policies with PPO-Lag and PPO to construct the `safe-high` and `unsafe-high` datasets, respectively.

To obtain the `safe-medium` and `unsafe-medium` datasets, we identified policies whose returns were roughly half of those of the corresponding near-optimal policies. For `safe-medium`, we ensured that the selected policy continued to satisfy the prescribed cost constraints, whereas the `unsafe-medium` policies intentionally violated them.

We describe policies for collecting each dataset as follows:

- `safe-high`: Near-optimal policy for the CMDP.
- `unsafe-high`: Near-optimal policy for the unconstrained MDP.
- `safe-medium`: A policy with medium performance that adheres to cost constraints.
- `unsafe-medium`: A policy with medium performance that disregards cost constraints.

Task	Dataset	Safe Ratio	Performant Ratio	Safe-Performant Ratio	Num of traj.
Safe Navigation Tasks					
Safety PointCircle1	safe-replay	1.00	0.23	0.23	466
	performance-replay	0.49	1.00	0.49	219
	replay	0.56	0.32	0.21	1020
Safety PointCircle2	safe-replay	1.00	0.15	0.15	1069
	performance-replay	0.68	1.00	0.68	237
	replay	0.76	0.25	0.20	1620
Safe Velocity Tasks					
Safety HopperVelocity	safe-replay	1.00	0.16	0.16	1683
	performance-replay	0.54	1.00	0.54	502
	replay	0.65	0.26	0.18	3020
Safety Walker2dVelocity	safe-replay	1.00	0.15	0.15	1944
	performance-replay	0.40	1.00	0.40	727
	replay	0.64	0.29	0.17	3510
Safe HalfCheetahVelocity	safe-replay	1.00	0.16	0.16	2402
	performance-replay	0.31	1.00	0.31	1262
	replay	0.58	0.34	0.16	4820
Safety AntVelocity	safe-replay	1.00	0.14	0.14	2844
	performance-replay	0.40	1.00	0.40	973
	replay	0.66	0.28	0.16	4910
Safety HumanoidVelocity	safe-replay	1.00	0.10	0.10	8067
	performance-replay	0.42	1.00	0.42	1900
	replay	0.73	0.22	0.13	12 160
Safe Manipulation Tasks					
ShadowHand OverSafeJoint	safe-replay	1.00	0.09	0.09	24 292
	performance-replay	0.67	1.00	0.67	2281
	replay	0.93	0.20	0.16	28 550
ShadowHand OverSafeFinger	safe-replay	1.00	0.06	0.06	24 741
	performance-replay	0.65	1.00	0.65	2100
	replay	0.90	0.12	0.09	29 050

Table H: Statistics for three types of replay datasets for each task in the Safety Gymnasium.

Task	Dataset	Return	Feasible Return	Cost
Safe Navigation Tasks				
Safety PointCircle1	safe-high	41.57 ± 0.04	41.57 ± 0.04	2.75 ± 0.14
	unsafe-high	55.49 ± 0.08	11.02 ± 0.16	206.57 ± 0.84
	safe-medium	26.10 ± 0.06	26.10 ± 0.06	0.07 ± 0.04
	unsafe-medium	30.69 ± 0.47	9.45 ± 0.16	259.65 ± 2.78
Safety PointCircle2	safe-high	41.51 ± 0.03	41.51 ± 0.03	4.95 ± 0.19
	unsafe-high	55.50 ± 0.07	7.37 ± 0.08	399.14 ± 0.98
	safe-medium	26.08 ± 0.06	26.08 ± 0.06	0.02 ± 0.02
	unsafe-medium	31.65 ± 0.40	6.64 ± 0.12	352.09 ± 2.84
Safe Velocity Tasks				
Safety HopperVelocity	safe-high	1679.44 ± 3.15	1679.44 ± 3.15	9.81 ± 0.15
	unsafe-high	2458.49 ± 27.77	70.94 ± 0.04	586.80 ± 7.11
	safe-medium	1043.24 ± 15.11	1043.24 ± 15.11	6.08 ± 0.12
	unsafe-medium	1209.56 ± 3.74	59.71 ± 0.03	296.61 ± 0.91
Safety Walker2dVelocity	safe-high	3014.33 ± 3.47	3014.33 ± 3.47	8.48 ± 0.22
	unsafe-high	6578.93 ± 18.77	133.56 ± 0.07	964.04 ± 2.59
	safe-medium	978.51 ± 4.76	978.51 ± 4.76	18.13 ± 0.09
	unsafe-medium	2665.78 ± 48.18	278.77 ± 1.10	513.79 ± 9.77
Safety HalfCheetahVelocity	safe-high	3004.71 ± 0.20	3004.71 ± 0.20	10.76 ± 0.14
	unsafe-high	6889.71 ± 39.42	127.71 ± 39.42	930.27 ± 5.04
	safe-medium	2110.94 ± 7.71	2110.94 ± 7.71	20.20 ± 0.12
	unsafe-medium	3311.35 ± 13.67	158.62 ± 0.63	680.09 ± 3.56
Safety AntVelocity	safe-high	3336.91 ± 17.39	3336.91 ± 17.39	9.75 ± 0.16
	unsafe-high	5950.40 ± 37.57	126.10 ± 0.49	930.42 ± 5.89
	safe-medium	1698.48 ± 8.55	1698.48 ± 8.55	17.25 ± 0.17
	unsafe-medium	2375.21 ± 18.44	197.90 ± 2.15	330.67 ± 3.11
Safety HumanoidVelocity	safe-high	6680.07 ± 0.06	6680.07 ± 0.06	14.86 ± 0.22
	unsafe-high	11 553.07 ± 56.35	326.09 ± 0.46	943.91 ± 4.52
	safe-medium	3079.13 ± 53.70	3079.13 ± 53.70	7.06 ± 0.26
	unsafe-medium	5849.70 ± 73.63	459.06 ± 2.11	688.90 ± 9.04
Safe Manipulation Tasks				
ShadowHand OverSafeJoint	safe-high	17.69 ± 0.12	17.69 ± 0.12	23.50 ± 0.16
	unsafe-high	24.24 ± 0.09	11.89 ± 0.03	65.69 ± 0.11
	safe-medium	7.90 ± 0.08	7.90 ± 0.08	24.04 ± 0.18
	unsafe-medium	13.40 ± 0.09	10.86 ± 0.06	44.82 ± 0.12
ShadowHand OverSafeFinger	safe-high	19.12 ± 0.12	19.12 ± 0.12	25.17 ± 0.12
	unsafe-high	23.26 ± 0.10	10.67 ± 0.02	70.63 ± 0.09
	safe-medium	5.03 ± 0.05	5.03 ± 0.05	24.28 ± 0.17
	unsafe-medium	11.48 ± 0.10	7.56 ± 0.03	54.98 ± 0.18

Table I: Statistics of four controlled datasets for each task in the Safety Gymnasium benchmark (unnormalized, values are mean ± standard error).

G Details of Realistic Experiments

G.1 Hyperparameters

Hyperparameters	BC	FBC	JFBC	DemoDICE-S	DemoDICE-U	DWBC-S	SpoilDICE (Ours)	SafeDICE	COptiDICE
Policy distribution	Normal								
Batch size	512								
Discount factor γ	0.99								
Network sizes (π, ν, C_P, C_S)	[256, 256] (Safe Navigation/Velocity, DSRL tasks) [1024, 1024, 512] (Safe Manipulation tasks)								
Training iteration	1,000,000								
Evaluation frequency	10,000								
Algorithm specific configurations									
π learning rate	3×10^{-5}								
α (regularization coefficient)	-	-	-	0.1	0.1	7.5	-	(auto)	0.01
η (class prior)	-	-	-	-	-	0.5	-	-	-
κ update frequency	-	100	100	-	-	-	100	-	-
ν learning rate	-	-	-	3×10^{-5}	3×10^{-5}	-	3×10^{-5}	3×10^{-5}	3×10^{-5}
ν gradient penalty coefficient	-	-	-	10	10	-	10	10	10
C_P gradient penalty coefficient	-	-	0	10	10	-	10	10	10
C_P learning rate	-	-	3×10^{-5}	3×10^{-5}	3×10^{-5}	3×10^{-5}	3×10^{-5}	3×10^{-5}	3×10^{-5}
C_S gradient penalty coefficient	-	0	0	-	-	-	0	-	-
C_S learning rate	-	3×10^{-5}	3×10^{-5}	-	-	-	3×10^{-5}	-	-

Table J: Hyperparameters of each method for Safety Gynmasium tasks. For DSRL dataset, we report the best-performing result of SpoilDICE over $\kappa \in [0.001, 0.005, 0.01, 0.05, 0.1, 0.2]$ for each task. For replay and controlled datasets, we use $\kappa = 0.01$ without search.

G.2 Performance Comparison on Replay Datasets

In addition to DSRL dataset discussed in Section 5.2.1, we evaluate methods under a realistic data collection scenario in which $\mathcal{D}_\mathrm{P}$ and $\mathcal{D}_\mathrm{S}$ are collected separately. Specifically, we construct *replay datasets* by collecting a trajectory-level buffer during PPO-Lag (Ray et al., 2019) training and taking subsets of full trajectories from the replay buffer: $\mathcal{D}_\mathrm{P}$ consists of trajectories with performance above 90% of the safe-optimal return, while $\mathcal{D}_\mathrm{S}$ contains trajectories that satisfy the cost upper bounds.

Task	Metric	BC	FBC	JFBC	DemoDICE-S	DemoDICE-U	SpoilDICE (Ours)
SafetyPointCircle1	Feasible return	0.73 ± 0.05	**0.72 ± 0.06**	0.78 ± 0.08	**0.84 ± 0.02**	**0.89 ± 0.01**	**0.84 ± 0.03**
	Episode cost	1.68 ± 0.40	**0.88 ± 0.58**	2.42 ± 1.00	**0.59 ± 0.19**	**0.43 ± 0.13**	**0.58 ± 0.34**
SafetyPointCircle2	Feasible return	0.58 ± 0.02	0.64 ± 0.03	0.55 ± 0.05	**0.83 ± 0.03**	**0.83 ± 0.01**	**0.84 ± 0.01**
	Episode cost	2.13 ± 0.72	1.01 ± 0.22	4.93 ± 1.51	**0.70 ± 0.29**	**0.94 ± 0.24**	**0.12 ± 0.07**
SafetyHopperVelocity	Feasible return	**0.86 ± 0.08**	**0.83 ± 0.07**	**0.95 ± 0.01**	**0.86 ± 0.04**	**0.95 ± 0.01**	**0.94 ± 0.01**
	Episode cost	**0.69 ± 0.16**	**0.65 ± 0.12**	**0.30 ± 0.07**	**0.44 ± 0.13**	**0.30 ± 0.08**	**0.41 ± 0.14**
SafetyWalker2dVelocity	Feasible return	**0.57 ± 0.05**	**0.19 ± 0.04**	**0.66 ± 0.10**	**0.67 ± 0.03**	**0.74 ± 0.04**	**0.79 ± 0.02**
	Episode cost	**0.76 ± 0.08**	**0.67 ± 0.20**	**0.62 ± 0.10**	**0.68 ± 0.08**	**0.96 ± 0.03**	**0.52 ± 0.05**
SafetyHalfCheetahVelocity	Feasible return	**0.85 ± 0.04**	**0.86 ± 0.05**	**0.73 ± 0.11**	0.31 ± 0.05	0.76 ± 0.10	**0.96 ± 0.01**
	Episode cost	**0.42 ± 0.08**	**0.24 ± 0.03**	**0.95 ± 0.25**	8.04 ± 2.29	1.53 ± 0.35	**0.56 ± 0.13**
SafetyAntVelocity	Feasible return	**0.91 ± 0.01**	**0.78 ± 0.05**	**0.94 ± 0.02**	**0.97 ± 0.00**	**0.97 ± 0.00**	**0.98 ± 0.00**
	Episode cost	**0.57 ± 0.03**	**0.38 ± 0.04**	**0.47 ± 0.07**	**0.28 ± 0.04**	**0.55 ± 0.08**	**0.38 ± 0.07**
SafetyHumanoidVelocity	Feasible return	0.30 ± 0.02	0.26 ± 0.02	0.40 ± 0.04	0.36 ± 0.02	0.39 ± 0.05	0.42 ± 0.03
	Episode cost	6.65 ± 0.70	6.27 ± 0.61	4.49 ± 0.62	5.25 ± 0.50	6.50 ± 0.86	4.39 ± 0.70
ShadowHandOverSafeJoint	Feasible return	0.07 ± 0.01	0.06 ± 0.01	0.32 ± 0.01	0.15 ± 0.02	0.25 ± 0.01	0.62 ± 0.02
	Episode cost	1.18 ± 0.02	1.14 ± 0.02	1.09 ± 0.03	1.11 ± 0.02	1.17 ± 0.04	1.12 ± 0.01
ShadowHandOverSafeFinger	Feasible return	0.05 ± 0.00	0.04 ± 0.01	**0.44 ± 0.02**	0.12 ± 0.01	**0.37 ± 0.02**	**0.69 ± 0.02**
	Episode cost	1.16 ± 0.02	1.16 ± 0.03	**0.79 ± 0.03**	1.23 ± 0.02	**0.82 ± 0.03**	**0.80 ± 0.02**

Table K: Safety–Performance comparison on Safety Gymnasium tasks under the replay dataset. Results are averaged over five seeds with standard errors. Feasible return is normalized between 0 (random policy) and 1 (safe high-performing policy); cost is normalized between 0 and 1 (cost upper bound). **Bold** marks methods satisfying constraints (mean cost $\leq$ cost upper bound). Blue marks the best feasible return, and **bold-and-blue** indicates the best feasible return while satisfying constraints.

Results in Table K show that SpoilDICE consistently outperforms or matches baselines in feasible return while maintaining low costs. In locomotion tasks such as SafetyWalker2dVelocity, SafetyHalfCheetahVelocity, and SafetyAntVelocity, it achieves near-optimal returns with significantly fewer cost violations. On high-dimensional control (SafetyHumanoidVelocity), SpoilDICE performs comparably to the best baselines, while in manipulation tasks (ShadowHandOverSafeJoint, ShadowHandOverSafeFinger) it yields clearly higher feasible returns under comparable costs. These results demonstrate that SpoilDICE is effective in realistic data collection settings, where existing offline IL methods either sacrifice performance or incur excessive safety violations.

G.3 Performance Comparison on Controlled Datasets

Task	Metric	BC	FBC	JFBC	DemoDICE-S	DemoDICE-U	SpoilDICE (Ours)
SafetyPointCircle1	Feasible return	0.70 ± 0.05	**0.82 ± 0.07**	0.83 ± 0.02	**0.81 ± 0.02**	0.70 ± 0.04	**0.93 ± 0.01**
	Episode cost	2.14 ± 0.30	**0.68 ± 0.37**	1.20 ± 0.21	**0.47 ± 0.10**	2.99 ± 0.50	**0.10 ± 0.08**
SafetyPointCircle2	Feasible return	0.15 ± 0.04	0.14 ± 0.08	0.54 ± 0.03	0.59 ± 0.05	0.59 ± 0.01	0.84 ± 0.02
	Episode cost	13.64 ± 0.66	6.87 ± 2.85	7.92 ± 0.44	2.40 ± 0.58	7.36 ± 0.30	1.39 ± 0.35
SafetyHopperVelocity	Feasible return	**0.98 ± 0.00**	**0.80 ± 0.10**	**0.92 ± 0.03**	**0.68 ± 0.07**	**0.83 ± 0.05**	**0.84 ± 0.07**
	Episode cost	**0.56 ± 0.07**	**0.73 ± 0.08**	**0.49 ± 0.09**	**0.37 ± 0.03**	**0.57 ± 0.10**	**0.38 ± 0.09**
SafetyWalker2dVelocity	Feasible return	**0.10 ± 0.01**	**0.01 ± 0.01**	0.14 ± 0.02	0.22 ± 0.02	0.35 ± 0.08	**0.88 ± 0.05**
	Episode cost	**0.83 ± 0.25**	**0.07 ± 0.02**	2.94 ± 0.54	1.04 ± 0.08	4.30 ± 0.65	**0.83 ± 0.37**
SafetyHalfCheetahVelocity	Feasible return	0.13 ± 0.00	0.84 ± 0.10	0.23 ± 0.03	0.33 ± 0.02	0.19 ± 0.01	**0.92 ± 0.07**
	Episode cost	27.23 ± 1.32	1.20 ± 0.49	24.52 ± 1.96	7.16 ± 1.02	18.50 ± 1.98	**0.56 ± 0.31**
SafetyAntVelocity	Feasible return	0.06 ± 0.00	**0.54 ± 0.09**	0.15 ± 0.02	**0.60 ± 0.00**	0.11 ± 0.01	**0.74 ± 0.02**
	Episode cost	34.67 ± 0.55	**0.27 ± 0.03**	31.57 ± 1.46	**0.60 ± 0.03**	31.77 ± 0.63	**0.11 ± 0.01**
SafetyHumanoidVelocity	Feasible return	0.03 ± 0.00	0.04 ± 0.01	0.03 ± 0.00	**0.21 ± 0.01**	0.03 ± 0.00	**0.30 ± 0.03**
	Episode cost	3.63 ± 0.10	1.12 ± 0.12	4.98 ± 0.26	**0.26 ± 0.04**	11.50 ± 0.00	**0.63 ± 0.11**
ShadowHandOverSafeJoint	Feasible return	0.25 ± 0.02	**0.44 ± 0.03**	0.31 ± 0.02	**0.38 ± 0.00**	0.27 ± 0.04	**1.09 ± 0.02**
	Episode cost	1.38 ± 0.03	**0.80 ± 0.00**	1.37 ± 0.05	**0.99 ± 0.06**	1.39 ± 0.06	**0.75 ± 0.02**
ShadowHandOverSafeFinger	Feasible return	0.11 ± 0.00	0.10 ± 0.00	0.11 ± 0.00	0.09 ± 0.00	0.11 ± 0.00	0.52 ± 0.01
	Episode cost	1.68 ± 0.01	1.61 ± 0.02	1.73 ± 0.02	1.56 ± 0.01	1.74 ± 0.01	1.66 ± 0.01

Table L: Comparison across nine Safety Gymnasium tasks on controlled datasets with balanced numbers of $\mathcal{D}_{\mathrm{P}}$ and $\mathcal{D}_{\mathrm{S}}$. For six tasks in the upper block, $|\mathcal{D}_{\mathrm{P}}| = |\mathcal{D}_{\mathrm{S}}| = 10$ is used, and $200, 1000, 1000$ trajectories are used for Humanoid, HandOverSafeJoint, HandOverSafeFinger tasks respectively. Results are averaged over 5 random seeds with standard errors reported. Feasible return is normalized between 0 (random policy) and 1 (safe high-performing policy), while the cost is normalized between 0 and 1 (cost upper bound). (**Bold** indicates constraint-satisfying methods (average cost $\leq$ cost upper bound). Blue highlights the method achieving the highest feasible return. When a method achieves the highest feasible return while satisfying safe constraints, it is shown in **bolded blue**.)

G.4 Data Efficiency Analysis

While Section 5.2.2 with Figure 4b and Figure 4a presents the overall data efficiency analysis with respect to $|\mathcal{D}_S|$ and $|\mathcal{D}_P|$, this section extends those findings by reporting additional experiments that are not included in the main paper due to space limitations.

In Section G.4.1, we analyze data efficiency by varying the dataset size parameter k, where $|\mathcal{D}_S| = |\mathcal{D}_P| = 2k$. Specifically, we vary the number of demonstrations as $k \in \{1, 3, 5, 7, 10\}$ while keeping $|\mathcal{D}_U| = 2000$. Figure B presents the aggregated results across six benchmark tasks, providing a high-level comparison among the methods. Figure C reports the per-task results, illustrating how data efficiency differs across individual environments.

Section G.4.2 and Section G.4.3 provide more detailed experiments with controlled datasets. Figure F reports per-task analyses with respect to $|\mathcal{D}_S|$, keeping $|\mathcal{D}_P| = 10$ and $|\mathcal{D}_U| = 2000$ fixed. In contrast, Figure G presents per-task analyses focusing on the changes of $|\mathcal{D}_P|$, keeping $|\mathcal{D}_S| = 10$ and $|\mathcal{D}_U| = 2000$ fixed. These breakdowns complement the main results by highlighting how SpoilDICE and other methods exhibit performance–safety trade-offs under varying dataset sizes across different environments.

G.4.1 Data Efficiency with Joint Scaling

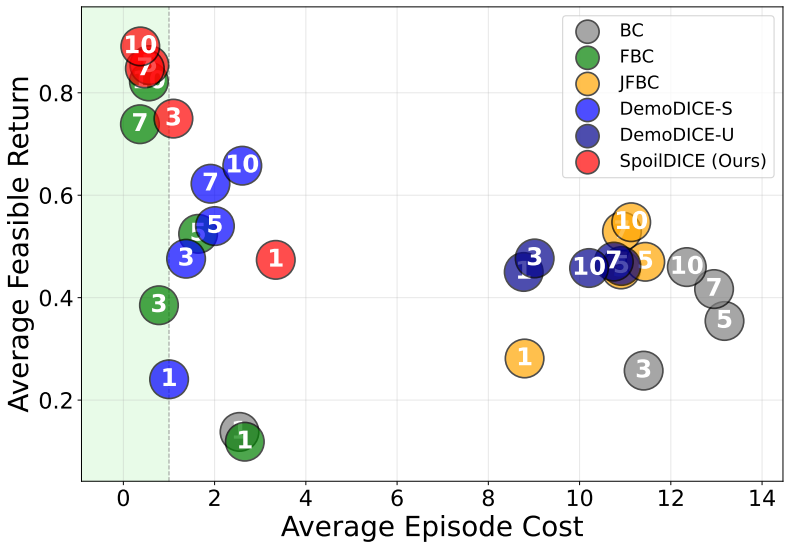


Figure B: Overall data efficiency averaged over six tasks of each method by jointly varying k, where $|\mathcal{D}_P| = |\mathcal{D}_S| = 2k$.

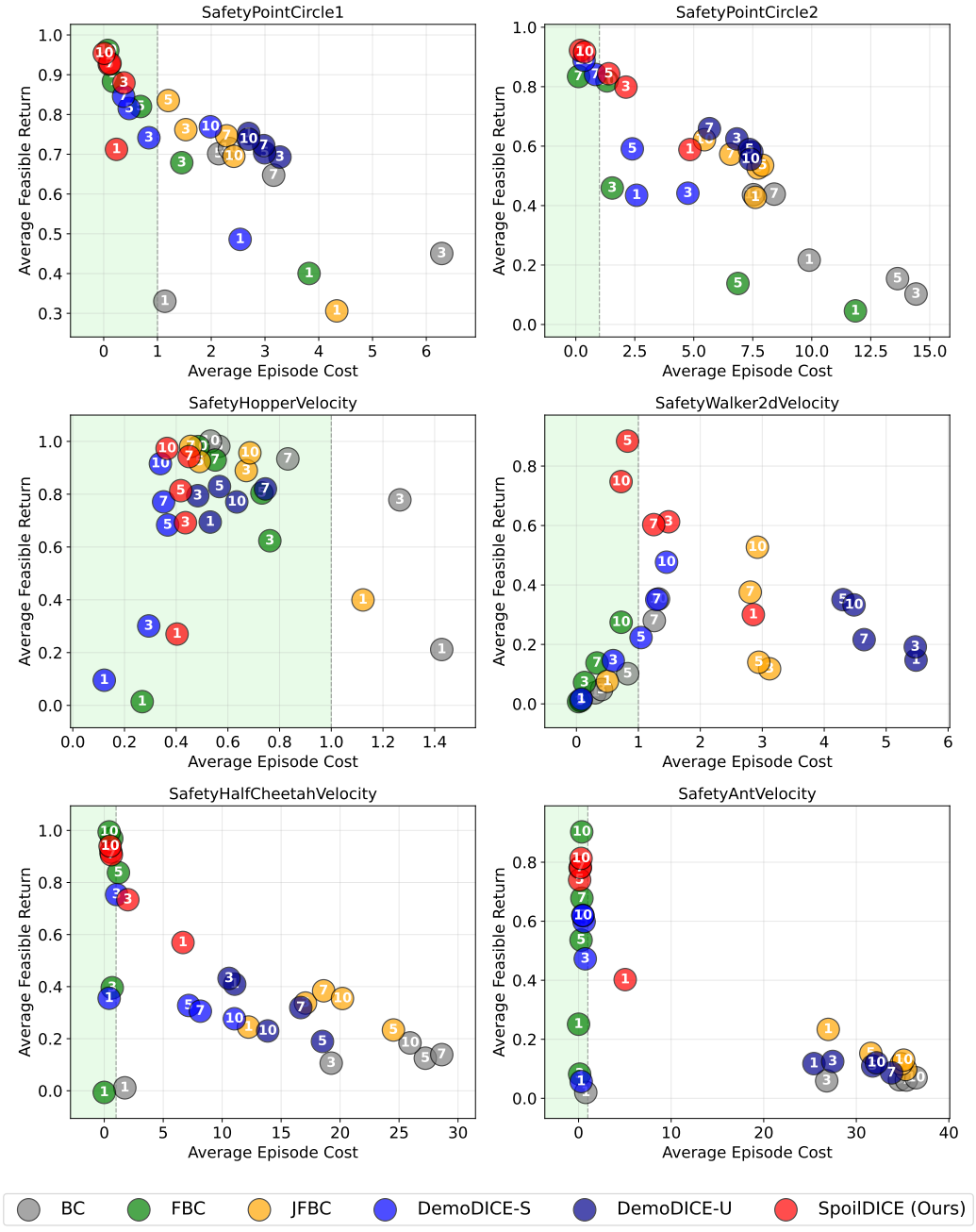


Figure C: Per-task analysis on data efficiency with jointly varying k where $|\mathcal{D}_{\mathrm{P}}| = |\mathcal{D}_{\mathrm{S}}| = 2k$.

G.4.2 Data Efficiency on $\mathcal{D}_S$

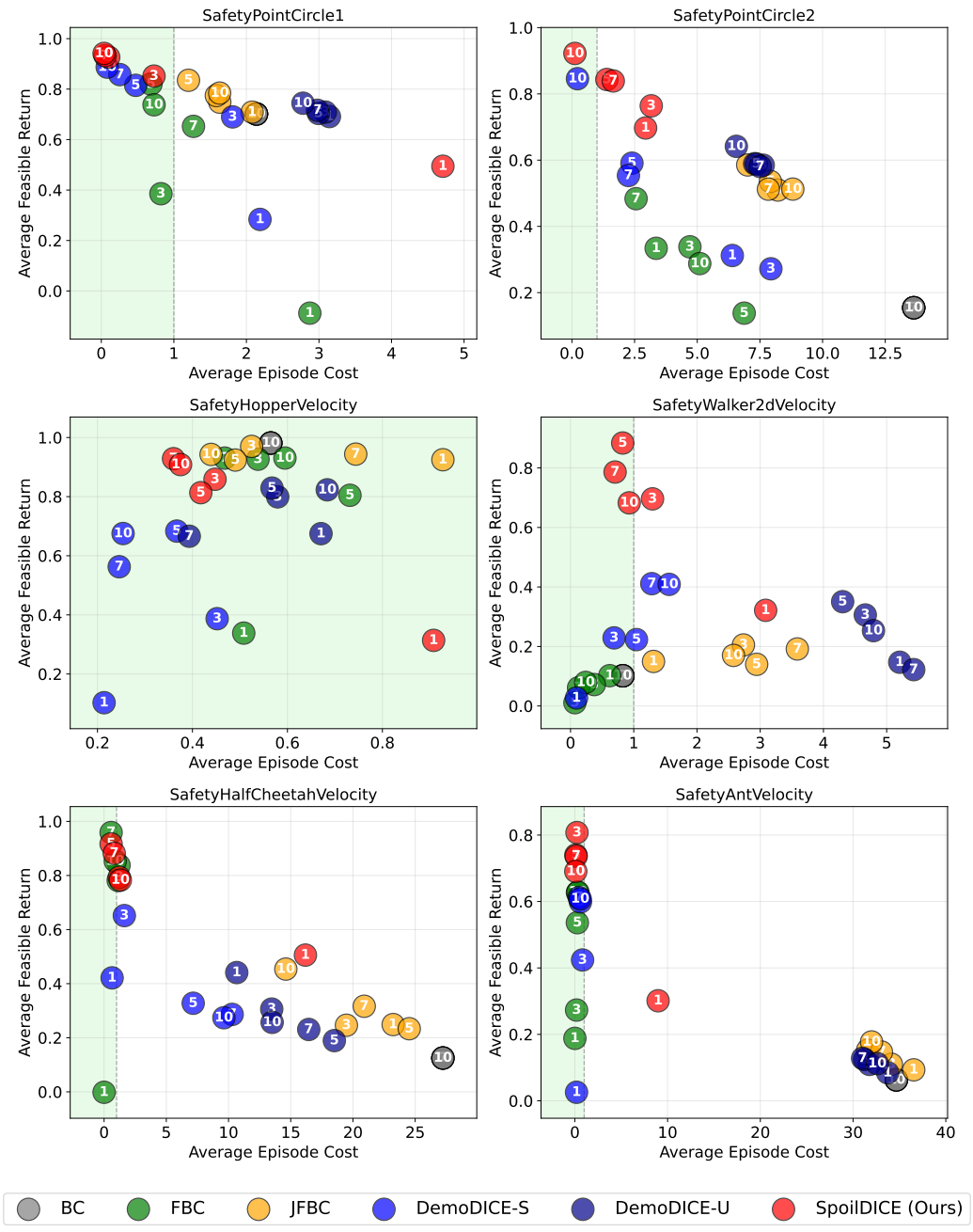


Figure D: Per-task analysis on data efficiency of safe demonstrations $\mathcal{D}_S$.

G.4.3 DATA EFFICIENCY ON $\mathcal{D}_{\mathrm{P}}$

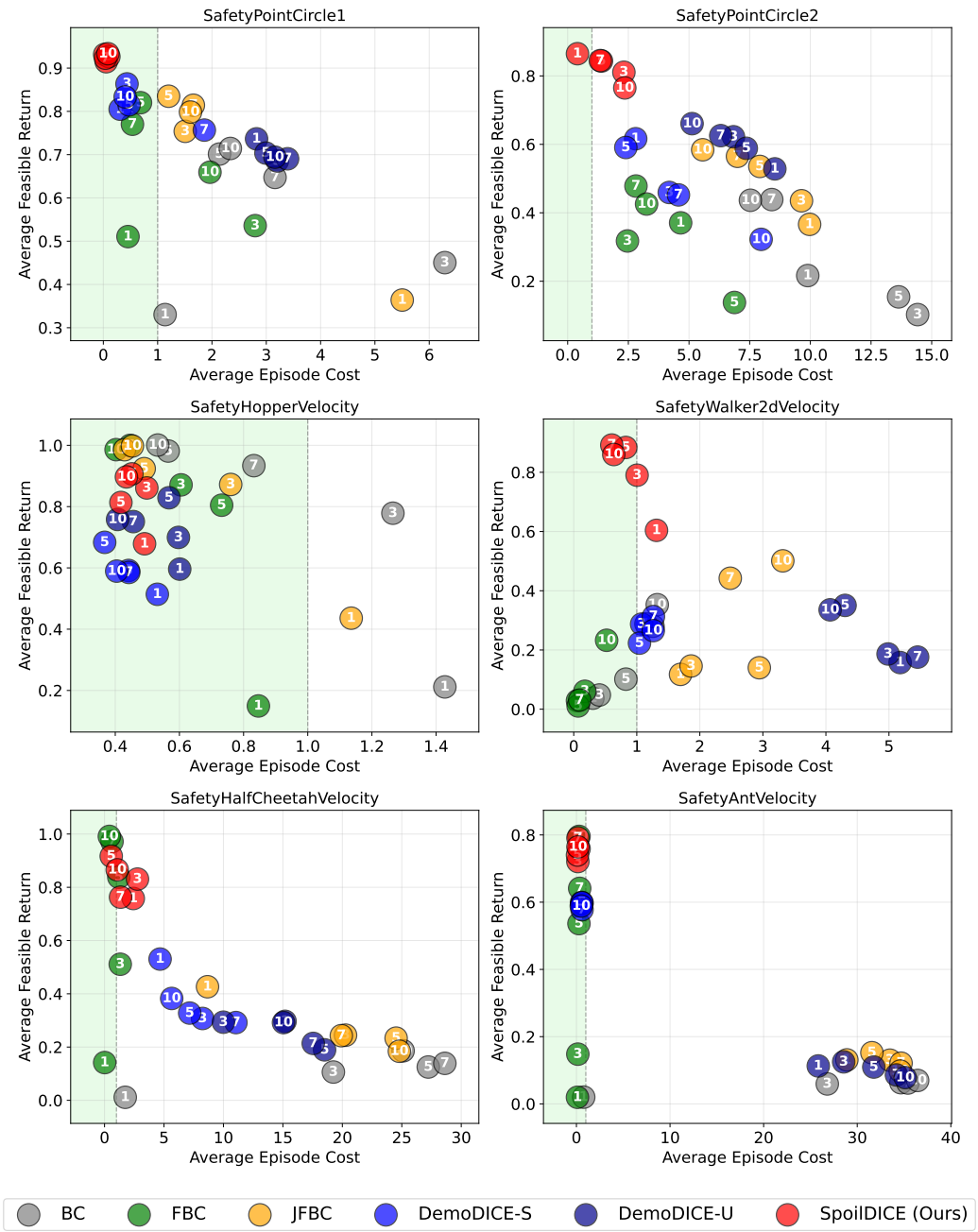


Figure E: Per-task analysis on data efficiency of performant demonstrations $\mathcal{D}_{\mathrm{P}}$.

G.5 Data Quality Analysis

In addition to the overall data quality analysis presented in Section 5.2.3 (Figure 4c and Figure 4d), we provide a task-specific analysis under a fixed dataset size of $|\mathcal{D}_{\mathrm{P}}| = |\mathcal{D}_{\mathrm{S}}| = 20$ and $|\mathcal{D}_{\mathrm{U}}| = 2000$. Figure G investigates the effect of varying **safeness**$(\mathcal{D}_{\mathrm{P}}) \in \{5\%, 25\%, 50\%, 75\%, 95\%\}$ while holding **performance**$(\mathcal{D}_{\mathrm{S}}) = 50\%$ constant. Conversely, Figure F illustrates how changes in **performance**$(\mathcal{D}_{\mathrm{S}})$ at the same scale influence the outcomes when **safeness**$(\mathcal{D}_{\mathrm{P}}) = 50\%$ is fixed.

G.5.1 Data Quality on $\mathcal{D}_{\mathrm{S}}$

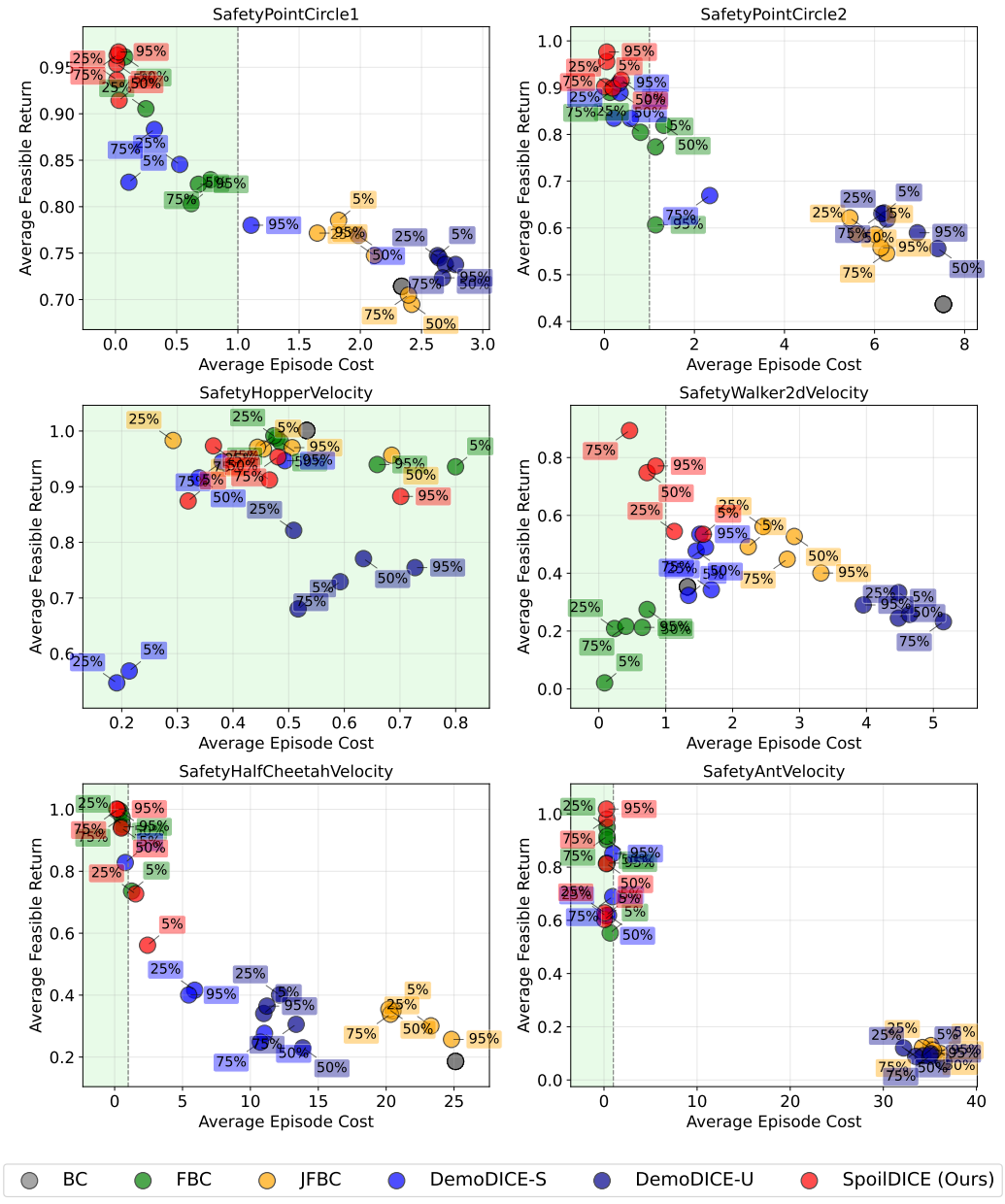


Figure F: Per-task performance for dependencies on performance of $\mathcal{D}_{\mathrm{S}}$.

G.5.2 Data Quality on $\mathcal{D}_{\mathrm{P}}$

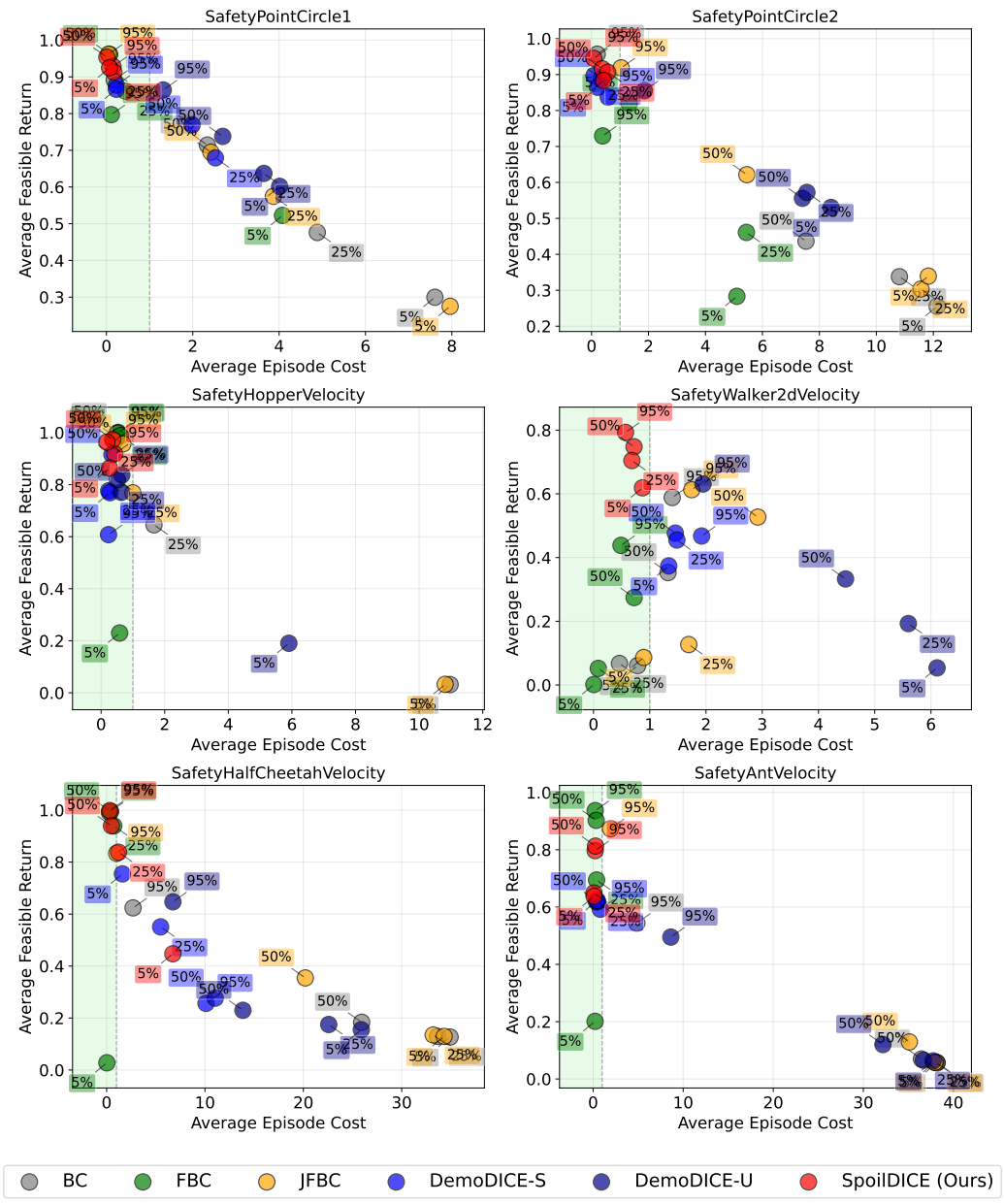


Figure G: Per-task performance analysis for dependencies on safeness of $\mathcal{D}_{\mathrm{P}}$.

G.6 Robustness on Noisy Safe Demonstrations

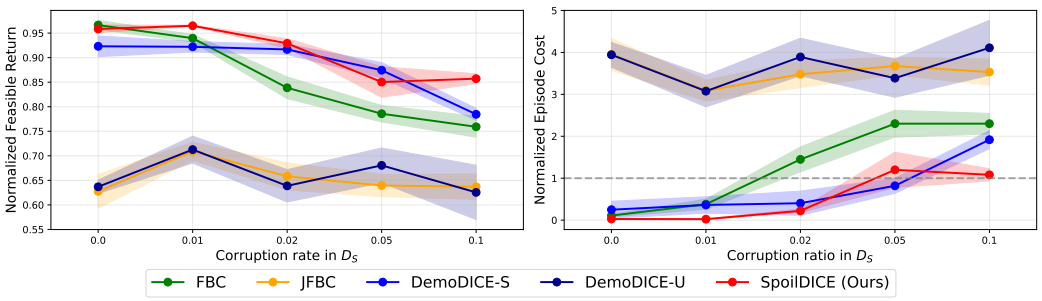


Figure H: Stability under corruption of safe demonstrations. We evaluate on the controlled dataset of the PointCircle1 task, using a sufficient number of demonstrations($|\mathcal{D}_{\mathrm{P}}| = 200$, $|\mathcal{D}_{\mathrm{S}}| = 200$). The x-axis indicates the corruption rate of $\mathcal{D}_{\mathrm{S}}$, i.e., the proportion of safe demonstrations replaced with unsafe demonstrations that exceed the episode cost upper bound. Both SpoilDICE and DemoDICE-S maintains relatively high feasible return and low episode cost even under moderate corruption, while other methods degrade significantly as $\mathcal{D}_{\mathrm{S}}$ becomes increasingly corrupted.

G.7 Sensitivity Analysis on Quantile Cutoff κ

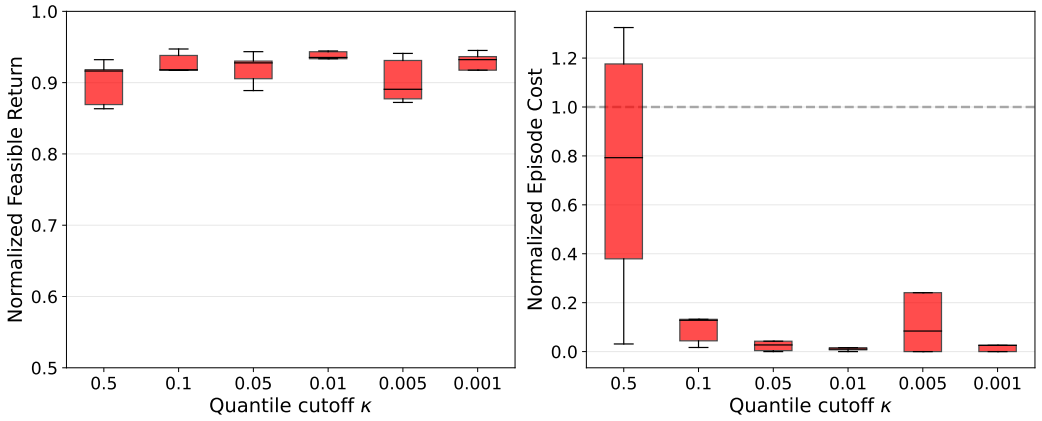


Figure I: Hyperparameter sensitivity analysis of SpoilDICE with respect to κ used for threshold of $\mathbb{1}_{d_{\mathrm{S}}}$. Experiments are conducted on controlled datasets of the PointCircle1 task, where $|\mathcal{D}_{\mathrm{P}}| = |\mathcal{D}_{\mathrm{S}}| = 10$, with 5 repetitions for each setting. SpoilDICE remains stable with respect to both feasible return and episode cost when the κ is set to a reasonable level (≤ 0.1).

H Disclosures

H.1 The Use of Large Language Models

We used ChatGPT to revise the draft; however, we emphasize that the core ideas of this work are not derived from large language models. We also used LLM agents during code implementation for debugging and automating repetitive tasks. In addition, the images in Figure 1 were edited also by ChatGPT for upscaling, color enhancement. The original images were captured from videos available on the CARLA simulator YouTube channel from the following links:

- `https://www.youtube.com/watch?v=S2VIP0qumas`
- `https://www.youtube.com/watch?v=J4pHRM1Q5nQ`