PSIL: LEARNING SAFE OFFLINE POLICY FROM PREFERRED AND NON-PREFERRED DEMONSTRATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Offline imitation learning (IL) typically relies on expert (i.e., preferred) demonstrations and a large collection of supplementary trajectories for broader state-action coverage. However, existing approaches often struggle to effectively integrate the valuable signal from non-preferred demonstrations, which provide critical information about what the agent should avoid. We propose PSIL, Progressive Safe Imitation Learning, a self-training-based offline IL algorithm that can flexibly learn from preferred demonstrations, non-preferred demonstrations, or both. PSIL progressively propagates behavioral information from this small set of demonstrations to a large supplementary trajectory dataset by assigning preference-weights based on their similarity to preferred or non-preferred behaviors. These weights induce a relative ordering over trajectories and are used in weighted behavior cloning to emphasize preferred behavior while down-weighting non-preferred behavior. We evaluate PSIL in Learning from Preferred (LfP), Learning from Non-preferred (LfN), and Learning from Preferred and Non-preferred (LfP+N) settings on the DSRL benchmark. Across these varied settings, PSIL maintains a favorable balance between reward and safety trade-off. We support our empirical results with theoretical and empirical analysis of the trajectory-weighting mechanism.

1 INTRODUCTION

Imitation learning (IL) provides an effective approach for training autonomous agents by learning from demonstrated behavior (Pomerleau, 1991; Zare et al., 2024). By leveraging expert or preferred demonstrations, IL methods have been successfully applied across a wide range of domains and can learn complex behaviors from relatively limited data (Kostrikov et al., 2020; Kim et al., 2022; Garg et al., 2021). However, standard IL primarily focuses on reproducing preferred behavior and provides little information to the agent about behaviors that should be avoided. This limitation is particularly important in safety-critical applications, where learning what *not* to do can be as important as learning what to do. We refer to the traditional setting in which only preferred demonstrations are available as *Learning from Preferred demonstrations (LfP)*. In this setting, the learner is encouraged to imitate preferred behavior without receiving explicit information about non-preferred behavior.

A more recent alternate line of work considers *Learning from Non-preferred demonstrations (LfN)*, where non-preferred trajectories are used to identify and avoid non-preferred behavior (Jang et al., 2023; Hoang et al., 2025; Burnwal et al., 2026b;a). Such trajectories can often be collected naturally as a byproduct of system deployment. For example, user-reported toxic conversations can provide examples of undesirable language behavior, while traffic violations such as running red lights or collisions can provide examples of unsafe driving. These non-preferred trajectories provide valuable information about behaviors that the learned policy should avoid. Nevertheless, LfN methods primarily exploit negative supervision and do not directly leverage the information contained in preferred demonstrations.

More generally, preferred and non-preferred demonstrations provide complementary forms of supervision: preferred demonstrations indicate *what the agent should do*, whereas non-preferred demonstrations indicate *what the agent should avoid*. In this work, we study the problem of learning from both types of supervision in the offline setting, where environment interaction is unavailable and the learner must exploit a fixed dataset. In particular, we consider a setting with a small number of preferred and/or non-preferred demonstrations together with a substantially larger collection of

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

supplementary trajectories containing diverse behaviors with varying levels of reward and cost. We refer to this larger collection as the *unlabeled trajectory dataset*.

We propose PSIL, *Progressive Safe Imitation Learning*, a self-training-based algorithm that learns from preferred, non-preferred, or both demonstrations. The key idea is to progressively propagate behavioral information from a small set of preferred or non-preferred demonstrations to the unlabeled trajectory dataset. At each iteration, PSIL identifies trajectories that exhibit behavior similar to preferred or non-preferred demonstrations and assigns them positive or negative decayed preference-weights, respectively. These weights induce a relative ordering over the unlabeled trajectory dataset, which is subsequently incorporated into a weighted behavior cloning objective. This allows trajectories that are more consistent with preferred behavior to have greater influence on policy learning while down-weighting trajectories with non-preferred behavior. Our contributions are:

1. **Progressive Safe Imitation Learning.** We propose PSIL, a self-training-based offline imitation learning algorithm that progressively identifies trajectories similar to preferred or non-preferred demonstrations and assigns them preference-weights. These weights are then used to learn a preferred policy through weighted behavior cloning. PSIL can operate under LfP, LfN, and LfP+N settings within a single framework.

2. **Analysis of progressive trajectory identification and regularization.** We provide an empirical analysis of the trajectory-weighting procedure. Additionally, we justify the score function design by outlining its Lipschitz continuity property and provide both theoretical and empirical evidence demonstrating that the regularization term prevents a constant-score solution.

3. **Empirical evaluation.** We evaluate PSIL across LfP, LfN, and LfP+N settings on the DSRL benchmark and compare it against seven state-of-the-art baselines. To evaluate, we introduce an Fβ-score, a weighted harmonic mean of normalized total reward and normalized safety, to jointly evaluate task performance and safety. While reward and cost returns are individually informative, evaluating them separately can make comparisons difficult when methods exhibit different reward-safety trade-offs. Across diverse Safety-Gym and Bullet-Gym tasks, PSIL maintains a favorable balance between reward-safety trade-off. We further analyze the sensitivity of PSIL to key hyperparameter and study the effect on its progressive trajectory-weighting mechanism.

2 Related Works

Offline Imitation Learning. Imitation learning (IL) learns policies by mimicking expert demonstrations. DAgger (Ross et al., 2011), GAIL (Ho & Ermon, 2016), AIRL (Fu et al., 2018), DAC (Kostrikov et al., 2019), and ValueDICE (Kostrikov et al., 2020) have demonstrated strong performance, but typically rely on online interaction to obtain samples from the learner's state-action distribution. Such interaction can be costly or infeasible in safety-critical settings. Offline IL instead learns from a fixed dataset without additional environment interaction. Methods such as DemoDICE (Kim et al., 2022), IQ-Learn (Garg et al., 2021), ILID (Yue et al., 2024), and IPL (Hejna & Sadigh, 2023) leverage additional offline trajectories beyond expert demonstrations to mitigate distribution shift and improve policy learning.

Learning from Non-Preferred Demonstrations. Recent work has investigated learning policies from non-preferred or negative demonstrations, which we refer to as *Learning from Non-Preferred demonstrations (LfN)*. SafeDICE (Jang et al., 2023) incorporates non-preferred trajectories into stationary-distribution correction to learn policies that balance task performance and safety. UNIQ (Hoang et al., 2025) extends the IQ-Learn framework (Garg et al., 2021) to learn policies from non-preferred demonstrations. OSIL (Burnwal et al., 2026a) learns a preference-based cost model from non-preferred trajectories and incorporates the learned cost into constrained policy optimization. SafeMIL (Burnwal et al., 2026b) formulates safe policy learning from non-preferred trajectories as a multiple-instance learning problem. These methods demonstrate the value of negative supervision for learning safer policies.

Prior methods primarily focus on exploiting either preferred demonstrations or non-preferred demonstrations. Recent works have also explored combining preferred and non-preferred feedbacks in *online* settings (Hoang et al., 2024; Abdolmaleki et al., 2025). In our work, we study the corresponding problem in an offline setting, where no further interaction is available. We propose PSIL, a unified algorithm that can use *preferred, non-preferred, or both demonstrations* to learn policy.

Self-Training. Self-training (Scudder, 1965a;b; Xie et al., 2020; Amini et al., 2025) is a semi-supervised learning paradigm that leverages a small set of labeled examples together with a larger pool of unlabeled data. The approach first trains a model using the available labeled data and then uses the model to generate predictions for the unlabeled examples. Predictions made with sufficiently high confidence are added to the labeled training set, after which the model is retrained on the expanded training set. This process is repeated, allowing the model to progressively exploit the information from the unlabeled data.

PSIL adopts the progressive expansion procedure of self-training to iteratively identify trajectories in the offline dataset that are similar to the preferred and non-preferred demonstrations. Rather than assigning discrete labels, PSIL assigns decayed preference-weights to newly identified trajectories according to their similarity confidence. Preferred trajectories receive positive weights, while non-preferred trajectories receive negative weights, with trajectories identified in earlier iterations receiving weights of larger magnitude than those incorporated in later iterations. These weights are then subsequently used in policy learning.

3 PRELIMINARIES

Constrained Markov Decision Process (CMDP). We consider a CMDP Altman (1999), represented by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, r, \rho_0, \gamma, \mathcal{C})$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, respectively. The transition dynamics are specified by $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the immediate reward function, $\rho_0 \in \Delta(\mathcal{S})$ is the initial state distribution, and $\gamma \in (0, 1)$ is the discount factor. The constraint tuple $\mathcal{C} = (c, b)$, where $c : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_{\geq 0}$ denotes the cost function and $b \in \mathbb{R}_{\geq 0}$ is the corresponding cost threshold. A policy $\pi : \mathcal{S} \to \mathcal{A}$ maps each state to an action. The preferred optimal policy $\pi^\star$ is obtained by maximizing the expected long-term discounted reward such that the expected long-term cost is bounded: $\max_\pi \mathbb{E}_{\tau\sim\pi}[\sum_t \gamma^t r(s_t, a_t)]$ s.t. $\mathbb{E}_{\tau\sim\pi}[\sum_t \gamma^t c(s_t, a_t)] \leq b$.

Imitation learning (IL) aims to learn a policy by mimicking preferred demonstrations, without requiring access to the underlying reward or cost functions. In this work, we consider the offline IL setting, where the policy is learned from a fixed set of pre-collected trajectories. These trajectories contain only state-action transitions and do not contain per-timestep reward and cost information, i.e., $\tau = (s_0, a_0, s_1, a_1, \ldots, s_H)$ of fixed horizon H. Let $\|\cdot\|$ denote the Frobenius norm. We define our dataset as follows:

Definition 1 (Preferred Trajectory Dataset)**.** *A preferred trajectory dataset ($\mathcal{D}_P$) consists of expert trajectories that achieve high reward while incurring low cost.*

Definition 2 (Non-Preferred Trajectory Dataset)**.** *A non-preferred trajectory dataset ($\mathcal{D}_N$) consists of trajectories that achieve low reward or incur high cost. Specifically, it includes trajectories that (i) achieve low reward and low cost, (ii) achieve low reward and high cost, or (iii) achieve high reward but incur high cost.*

Definition 3 (Unlabeled Trajectory Dataset)**.** *A unlabeled trajectory dataset ($\mathcal{D}_U$) consists of trajectories spanning a full range of behaviors, including preferred, non-preferred, and intermediate reward-cost trade-offs.*

While the above dataset definitions use reward and cost, these are only used to curate the datasets. The numerical reward and cost values are not available to the learning agent. We assume access to a small set of preferred trajectories, $\mathcal{D}_P$, a small set of non-preferred trajectories, $\mathcal{D}_N$, and a large collection of unlabeled trajectories, $\mathcal{D}_U$. Similar to standard offline IL, the preferred dataset $\mathcal{D}_P$ specifies the target behavior, while the unlabeled dataset $\mathcal{D}_U$ provides broader coverage of the state-action space, reducing the impact of distributional shift. In our work, we also use the non-preferred dataset $\mathcal{D}_N$ to explicitly encourage the learned policy to avoid unsafe or suboptimal actions.

4 METHODOLOGY

This section introduces self-training based PSIL, Progressive Safe Imitation Learning, algorithm. Starting from a small set of preferred ($\mathcal{D}_P$) and non-preferred ($\mathcal{D}_N$) demonstrations, PSIL algorithm progressively assigns decayed preference-weights to trajectories in the unlabeled dataset ($\mathcal{D}_U$) according to their resemblance to the preferred and non-preferred demonstrations. Trajectories identi-

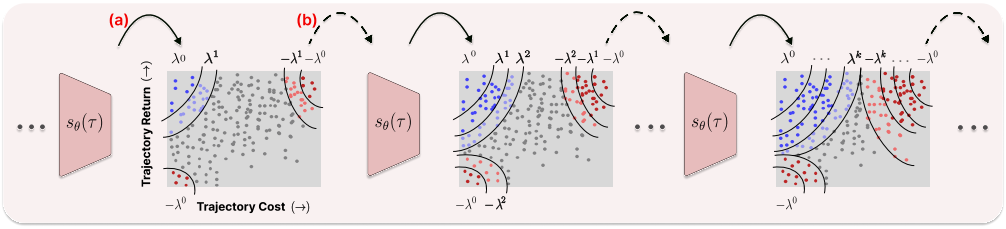

Figure 1: **Progressive Trajectory Weight Assignment.** Color encodes each trajectory's status: blue/red indicate similarity to preferred/non-preferred behavior (positive/negative weight), gray indicates not yet assigned. Light shades mark trajectories newly identified in the current iteration; dark shades mark trajectories identified earlier and already used to train score function, s_θ. (a) Solid arrows: At each trajectory-weighting step, s_θ scores each $\tau_U \in \mathcal{D}_U$ and assigns a weight decayed by factor of λ to newly identified trajectories evaluated as sufficiently similar. (b) Dashed arrows: s_θ is retrained on all identified trajectories so far. Retraining s_θ shifts the scores of nearby trajectories, letting the next iteration identify a wider set. As $\lambda \in (0, 1)$, weight magnitude decays, reflecting decreasing confidence as similarity propagates outward. Repeating (a)→(b) progressively expands supervision from $\mathcal{D}_P/\mathcal{D}_N$ across the full dataset.

fied in earlier iterations receive larger-magnitude weights, with positive weights assigned to trajectories resembling preferred demonstrations and negative weights assigned to those resembling non-preferred demonstrations. These weights induce a relative ordering over the unlabeled trajectory dataset and determine the contribution of each trajectory during policy learning. The policy is then learned using a weighted behavior cloning objective, in which trajectories with higher preference-weights contribute more strongly to the optimization objective. This encourages the learned policy to imitate preferred behaviors while still leveraging the entire offline dataset for training. The remainder of this section first presents the progressive trajectory-weighting procedure and then describes the weighted behavior cloning algorithm used to learn the preferred policy.

4.1 Trajectory Weight Assignment

Let $s_\theta : \tau \to [-1, 1]$ denote a score function parameterized by θ. For bounded trajectory inputs satisfying $\|\tau\| \leq l$, we assume that s_θ satisfies the following Lipschitz continuity property:

Property 1. *Let τ_1 and τ_2 be trajectory sequences with $\|\tau_1 - \tau_2\| \leq \epsilon$. Then the score function satisfies the following:*

$$|s_\theta(\tau_1) - s_\theta(\tau_2)| \leq K\epsilon \tag{1}$$

That is, the score function preserves local similarity in the trajectory space: trajectories that are close to one another receive similar scores. We discuss the choice of score function in Appendix E.

We aim to assign preference-weights $w(\tau)$ to all trajectories in $\mathcal{D}_U$ such that trajectories similar to the preferred dataset receive higher positive weights, and those similar to the non-preferred dataset receive higher negative weights. Initially, $w(\tau_P) = 1.0$, $\forall\, \tau_P \in \mathcal{D}_P$; $w(\tau_N) = -1.0$, $\forall\, \tau_N \in \mathcal{D}_N$; and $w(\tau_U) = 0.0\, \forall\, \tau_U \in \mathcal{D}_U$, since no information is available on the trajectories from the unlabeled dataset. We train s_θ by minimizing:

$$\min_\theta \sum_{\tau \in \mathcal{D}_P \cup \mathcal{D}_N \cup \mathcal{D}_U} \mathbb{1}_{w(\tau)>0}\,(1.0 - s_\theta(\tau))^2 \;+\; \mathbb{1}_{w(\tau)<0}\,(-1.0 - s_\theta(\tau))^2 + \underbrace{s_\theta(\tau_r)^2}_{\text{degeneracy regularizer}} \tag{2}$$

where $\mathbb{1}$ is an indicator function and τ_r is an element-wise convex combination of trajectories sampled from the unlabeled dataset. The regularization term encourages the score function to assign a value equal to zero in τ_r, preventing degenerate solutions in which the score function collapses to a constant prediction. A detailed discussion on the effect of regularization is in section 5.4.

During the first iteration, only the trajectories of $\mathcal{D}_P$ and $\mathcal{D}_N$ contribute to the loss because all the trajectories of $\mathcal{D}_U$ have zero weights. After the score function converges, Property 1 implies that trajectories that are close in the trajectory space receive similar scores. Consequently, those trajectories, $\tau_U \in \mathcal{D}_U$, that are similar to preferred demonstrations will have large positive scores,

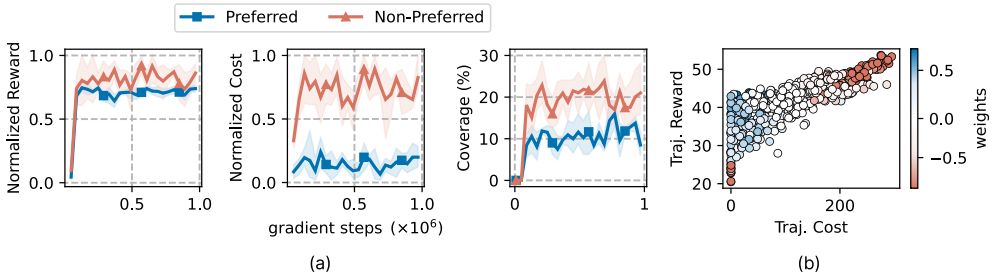

Figure 2: *Progressive Trajectory Weight Assignment on the PointCircle2 Environment.* (a) As training progresses, PSIL progressively identifies trajectories in the unlabeled dataset $\mathcal{D}_U$ that are similar to the preferred and non-preferred demonstrations. The identified preferred trajectories consistently exhibit high reward and low cost, whereas the identified non-preferred trajectories incur substantially higher cost. (b) Final trajectory weights after 1M training. Preferred trajectories with high reward and low cost receive large positive weights, while non-preferred trajectories with low reward or high cost receive large negative weights. These results demonstrate that the proposed progressive trajectory assignment successfully separates preferred and non-preferred behaviors.

while those similar to non-preferred demonstrations will have large negative scores. We select these trajectories and assign preference-weights:

$$w(\tau_U) \leftarrow \begin{cases} \lambda^k, & \text{if } w(\tau_U) = 0 \text{ and } s_\theta(\tau_U) > \eta \\ -\lambda^k, & \text{if } w(\tau_U) = 0 \text{ and } s_\theta(\tau_U) < -\eta \\ w(\tau_U), & \text{otherwise} \end{cases} \tag{3}$$

Here, $\eta \in (0, 1)$ is a confidence threshold on the score above which a trajectory is treated as similar to the preferred (or non-preferred) demonstrations, and $\lambda \in (0, 1)$ controls the magnitude of the assigned preference-weights. During the first iteration ($k = 1$), those trajectories whose scores exceed η (or fall below $-\eta$) receive preference-weights of λ (or $-\lambda$). We then retrain s_θ using equation 2, and the next set of identified trajectories are assigned weights of λ^2, followed by λ^3 in the next iteration, and so on. Since $\lambda < 1$, the magnitude of the assigned weights decays exponentially across iterations, reflecting the decreasing confidence in similarity with the preferred and non-preferred trajectories. Figure 1 summarizes how similarity information is progressively propagated from the preferred and non-preferred datasets to the trajectories in the unlabeled dataset through iterative preference-weight assignment and score function retraining.

The condition $w(\tau_U) = 0$ in equation 3 ensures that weights are assigned only to newly identified trajectories. Consequently, trajectories discovered in earlier iterations retain their larger-magnitude weights, reflecting higher confidence in their similarity to the preferred or non-preferred demonstrations. In contrast, trajectories identified in later iterations receive progressively smaller weights. The decayed weights are used only in the policy objective (section 4.2). When retraining s_θ, we regress the scores of newly identified trajectories toward ± 1, which allows each trajectory to push its neighbors' scores above η. Repeating this procedure gradually propagates similarity information from the preferred and non-preferred datasets to the unlabeled dataset.

Property 1 ensures that close trajectories receive similar scores, but it does not guarantee the converse. PSIL therefore treats s_θ as a learned proxy for similarity to the preferred and non-preferred demonstrations, and we empirically verify that the trajectories assigned positive (negative) weights exhibit preferred (non-preferred) behavior. Figure 2 provides this evidence and also shows that the proposed progressive trajectory assignment successfully expands the available supervision for policy learning. As shown in Figure 2(a), as training progresses, PSIL identifies an increasing number of trajectories in the unlabeled dataset that are similar to the preferred and non-preferred demonstrations. Although the identified non-preferred trajectories exhibit relatively high average reward, they consistently incur substantially higher cost. This behavior is expected because non-preferred trajectories are defined as those with either low reward or high cost, and in the PointCircle2 benchmark the majority belong to the latter category. Furthermore, Figure 2(b) shows that the propagation assigns large positive weights to low-cost, high-reward trajectories and large negative weights to trajectories with low reward or high cost. Appendix H report similar empirical evidence on other environments.

4.2 Learning Preferred Policy

For each trajectory in the unlabeled dataset, we have access to its corresponding preference-weights, which we can use to learn the preferred policy. Trajectories with larger positive weights are similar to the preferred demonstrations; therefore, imitating these trajectories encourages the learned policy toward preferred behavior. This observation leads to a weighted behavior cloning objective, where the imitation loss is weighted according to the preference-weight.

$$\min_{\pi} \mathbb{E}_{\tau \sim \mathcal{D}_U} \left[\exp\left(\frac{w(\tau)}{\alpha}\right) \sum_t \gamma^t \|a_t - \pi(s_t)\|_2^2 \right] \tag{4}$$

In this equation, $w(\tau)$ represents the preference-weight of the trajectory τ, and α is a temperature parameter that controls the sharpness of the weighting. The exponential transformation amplifies the influence of trajectories with positive weights and scales down the influence of trajectories with negative weights. Specifically, we use small temperature value to ensure that the negative weights are exponentially scaled down, thereby reducing the influence of non-preferred behavior during policy updates. As a result, the learned policy is encouraged to reproduce actions from trajectories that are more likely to exhibit the desired behavior, while still taking advantage of the entire offline dataset. Appendix B presents a computationally efficient implementation of PSIL.

5 Experiments

We evaluate PSIL on the Datasets for offline Safe RL (DSRL) benchmark (Liu et al., 2024) and compare it against the state-of-the-art offline imitation learning and safe imitation learning methods. We conduct experiments on six Safety-Gymnasium tasks (Point-Circle2, Car-Circle2, Point-Goal1, Car-Goal1, Ant-Velocity, and Swimmer-Velocity) and four Bullet-Gym tasks (Ant-Circle, Ant-Run, Drone-Circle, and Drone-Run), all from the official DSRL benchmark. For each task, we construct $\mathcal{D}_P$ using one preferred trajectory, $\mathcal{D}_N$ using ten non-preferred trajectories, and $\mathcal{D}_U$ using 500 trajectories randomly selected from the DSRL benchmark. This configuration is used throughout the paper unless stated otherwise (e.g., in ablation studies, where the size of $\mathcal{D}_N$is varied). To match the assumptions of our setting, all reward and cost annotations are removed from the offline dataset before training. Additional dataset details are in Appendix A.

Baselines. We evaluate PSIL in three learning settings and compare it against seven state-of-the-art baselines tailored to each setting.

Learning from Preferred and Non-preferred Demonstrations (LfP+N). We consider the setting where both preferred and non-preferred demonstrations ($\mathcal{D}_P$ and $\mathcal{D}_N$) are available in addition to the unlabeled dataset ($\mathcal{D}_U$). Since existing methods are not designed to exploit both sources of supervision simultaneously, we extend IPL (Hejna & Sadigh, 2023) and OSIL (Burnwal et al., 2026a) to utilize both preferred and non-preferred demonstrations, denoted as *IPL (LfP+N)* and *OSIL (LfP+N)*, respectively. Specifically, IPL is augmented to leverage non-preferred demonstrations during training, while OSIL is extended to additionally utilize preferred demonstrations. These modifications enable a fair comparison in the LfP+N setting, and implementation details are in Appendix D.

Learning from Preferred Demonstrations (LfP). This corresponds to the standard offline imitation learning setting, where expert demonstrations are regarded as preferred demonstrations. The agent has access only to the preferred demonstrations ($\mathcal{D}_P$) and the unlabeled offline dataset ($\mathcal{D}_U$). We compare PSIL with *BC* trained on the unlabeled dataset, *IQ-Learn* (Garg et al., 2021), *ILID* (Yue et al., 2024) and *IPL* (Hejna & Sadigh, 2023).

Learning from Non-preferred Demonstrations (LfN). Finally, in this setting, the agent has access only to non-preferred demonstrations ($\mathcal{D}_N$) and the unlabeled offline dataset ($\mathcal{D}_U$). We compare with *SafeDICE* (Jang et al., 2023), *UNIQ* (Hoang et al., 2025), and *OSIL* (Burnwal et al., 2026a).

Metrics. Following prior work (Liu et al., 2023; Jang et al., 2023; Hoang et al., 2025; Burnwal et al., 2026a), we evaluate each learned policy in terms of its total reward and cost. All policies are evaluated after 1M gradient update steps by averaging the total reward and cost over 10 evaluation trajectories generated by the learned policy. Each experiment is repeated with five independent random seeds. We report the mean performance across seeds, with 95% bootstrap confidence intervals computed from the seed-level results using 1,000 bootstrap samples. While total reward and

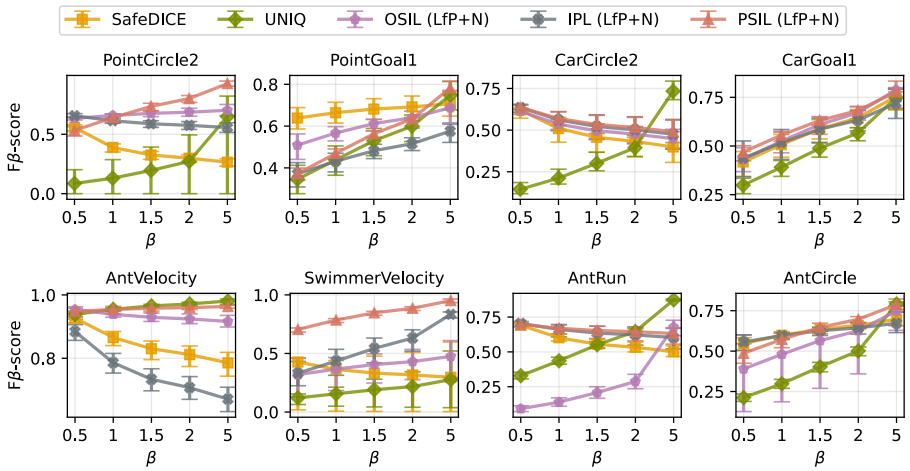

Figure 3: Comparison of the reward-safety trade-off in LfP+N setting. Each policy is trained for 1M gradient update steps and then evaluated using Fβ-score with $\beta \in \{0.5, 1, 1.5, 2, 5\}$. Smaller values of β place greater emphasis on reward, whereas larger values emphasize safety. PSIL maintains high Fβ-scores across different β values indicating better reward-safety trade-off. Error bars denote 95% bootstrap confidence intervals. SafeDICE and UNIQ uses only non-preferred demonstrations $\mathcal{D}_N$. Additional results are in Appendix I.

cost are informative, comparing algorithms using two separate metrics can be challenging when methods exhibit different reward-safety trade-offs. For example, a policy may achieve a very low total cost simply by sacrificing task performance, resulting in negligible reward. To summarize the reward–safety trade-off in a single number, we additionally report the Fβ-score, the weighted harmonic mean of normalized task performance and safety. The Fβ-score is commonly used in machine learning to evaluate a classification model's performance, where the β parameter can be explicitly adjusted to prioritize either precision or recall depending on the application context.

We first normalize the total reward using the task-specific minimum and maximum reward values provided by the DSRL benchmark:

$$R_{\text{norm}} = \frac{R - R_{\min}}{R_{\max} - R_{\min}} \tag{5}$$

We then define the normalized safety such that larger values correspond to safer policies (i.e., lower total cost), ensuring that both normalized reward and safety have the same interpretation and can be naturally combined using a weighted harmonic mean:

$$S_{\text{norm}} = \frac{C_{\max} - C}{C_{\max} - C_{\min}} \tag{6}$$

Here, $R_{\min}$, $R_{\max}$, $C_{\min}$, and $C_{\max}$ denote the task-specific reward and cost bounds provided by the DSRL benchmark. To ensure the normalized metrics remain strictly within $[0, 1]$, we clip the reward R to the range $[R_{\min}, R_{\max}]$ and the cost C to $[C_{\min}, C_{\max}]$. We then compute the Fβ-score as:

$$\text{F}\beta = \frac{(1+\beta^2)\, R_{\text{norm}}\, S_{\text{norm}}}{\beta^2 R_{\text{norm}} + S_{\text{norm}}} \tag{7}$$

The parameter β controls the relative importance of reward and safety when computing the Fβ-score. Specifically, $\beta < 1$ places greater emphasis on reward, $\beta = 1$ assigns equal importance to reward and safety, and $\beta > 1$ places greater emphasis on safety. Note that β is used only during evaluation and does not influence policy training; the same learned policy is evaluated using different values of β to assess its performance under varying reward-safety trade-offs. A truly safe high-reward policy should achieve consistently high Fβ-scores across a wide range of β values, demonstrating that it maintains strong task performance and safety regardless of the desired trade-off. For completeness, we also report the total reward and cost values in Appendix I, J, K.

5.1 Can PSIL Learn from Preferred and Non-preferred Demonstrations?

Figure 3 compares PSIL in the LfP+N setting using the Fβ-score. Across most benchmark tasks, PSIL achieves high Fβ-score over a wide range of β values, demonstrating that it learns policies that better balance task performance and safety.

For lower values of β, where reward is emphasized over safety, PSIL is competitive with the baseline methods. UNIQ in Point-Circle2, Car-Circle2, Car-Goal1 and OSIL in Ant-Run exhibit conservative behavior, learning policies with low reward and low cost. This is reflected by their lower Fβ-score when $\beta = 0.5$, where the evaluation places greater emphasis on reward. As β increases, these methods become increasingly competitive because the metric favors their low-cost behavior, occasionally outperforming PSIL for very large β.

By progressively exploiting both preferred and non-preferred demonstrations, PSIL learns policies that simultaneously maximize task performance while avoiding unsafe behaviors. Consequently, PSIL maintains a favorable performance across a broad range of reward-safety trade-offs.

5.2 Can PSIL Learn from Preferred Demonstrations?

Figure 4 reports the aggregated performance of all algorithms across the benchmark environments in the LfP setting, where the agent has access only to the preferred demonstrations ($\mathcal{D}_P$) and the unlabeled offline dataset ($\mathcal{D}_U$), without explicit supervision from non-preferred demonstrations. We report the stratified bootstrap aggregated Fβ-score in the main paper, while the per-task Fβ-scores and corresponding total reward and cost values are provided in Appendix J. PSIL achieves Fβ-scores comparable to IPL, the strongest baseline in this setting. These results indicate that PSIL can leverage the unlabeled offline dataset to obtain a competitive trade-off between task performance and safety using only preferred demonstrations.

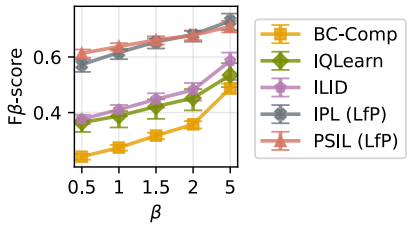

Figure 4: Aggregated performance for LfP setting. Each policy is trained for 1M gradient update steps. Smaller values of β place greater emphasis on reward, whereas larger values emphasize safety. Error bars: 95% bootstrap CI.

5.3 Can PSIL Learn from Non-Preferred Demonstrations?

In this setting, the agent has access only to the non-preferred demonstrations ($\mathcal{D}_N$) and the unlabeled dataset ($\mathcal{D}_U$). Figure 5 reports the stratified bootstrap aggregated performance, while Appendix K provides the per-task Fβ-scores and corresponding total reward and cost values. Despite having access only to non-preferred demonstrations, PSIL achieves competitive performance with existing LfN methods across the different values of β. These results indicate that PSIL progressively assigning lower similarity weights to trajectories that resemble the non-preferred demonstrations effectively propagates negative preference information throughout the unlabeled dataset, $\mathcal{D}_U$. Overall, the results indicate that PSIL can obtain a competitive reward-safety trade-off even without access to preferred demonstrations.

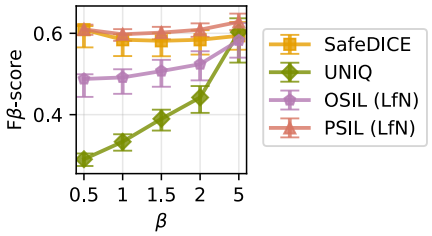

Figure 5: Aggregated performance for LfN setting. Each policy is trained for 1M gradient update steps. Smaller values of β place greater emphasis on reward, whereas larger values emphasize safety. Error bars: 95% bootstrap CI.

5.4 Effect of Regularization in Trajectory Weight Assignment

The regularization term in equation 2 plays a critical role in preventing degenerate solutions during trajectory weight assignment. This issue is particularly pronounced in the LfP and LfN settings, where supervision is available from only one type of demonstration. Consider the LfP setting where only preferred and unlabeled trajectory dataset is available. Under the LfP setting, if the regulariza-

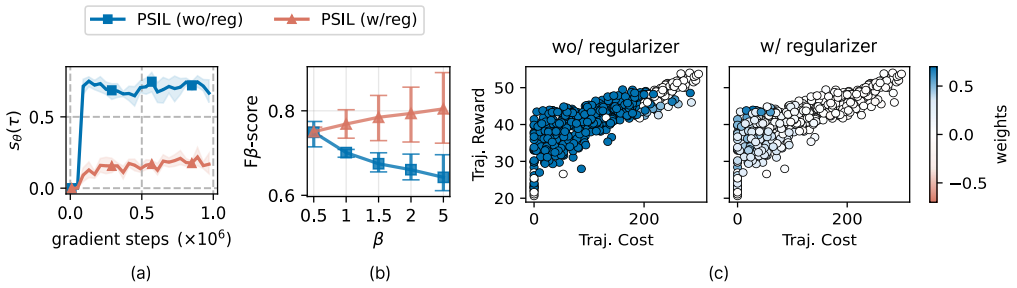

Figure 6: *Effect of regularization in the LfP setting for PointCircle2 environment.* (a) Trajectory score $s_\theta(\tau)$ during training with and without regularization. Without regularization, the score model collapses to a nearly constant prediction for all trajectories. (b) Removing the regularization term results in lower Fβ-scores. (c) Final trajectory weights after training. Without regularization, the collapsed score model assigns large positive weights to almost all trajectories. In contrast, the proposed regularization learns discriminative trajectory scores, resulting in accurate trajectory weight assignment and effective separation of preferred and non-preferred behaviors. Error bars denote 95% bootstrap confidence intervals.

tion term is removed then the score function optimization objective becomes:

$$\begin{aligned} &\min_\theta \sum_{\tau \in \mathcal{D}_P \cup \mathcal{D}_U} \mathbb{1}_{w(\tau)>0} \left(1.0 - s_\theta(\tau)\right)^2 + \underbrace{\mathbb{1}_{w(\tau)<0} \left(-1.0 - s_\theta(\tau)\right)^2}_{=0} + \underbrace{s_\theta(\tau_r)^2}_{=0} \\ &= \min_\theta \sum_{\tau \in \mathcal{D}_P \cup \mathcal{D}_U} \mathbb{1}_{w(\tau)>0} \left(1.0 - s_\theta(\tau)\right)^2 \end{aligned} \tag{8}$$

The second term in the objective, $\mathbb{1}_{w(\tau)<0} \left(-1.0 - s_\theta(\tau)\right)^2$, does not contribute to the optimization problem as there are no non-preferred trajectories and therefore there are no trajectories whose preference-weights are negative. Without additional constraints, the score function can minimize the training objective by assigning nearly identical score of 1.0 to every trajectory, resulting in poor preferred trajectory identification, and thereby poor policy learning. Figure 6 empirically demonstrates this phenomenon. Removing the regularization term for the LfP setting causes the score function to converge to a nearly constant value, leading to poor trajectory weight assignment and consequently inferior policy performance. In contrast, incorporating regularization term produces non-degenerate score function which leads to better trajectory weight assignment and better policy.

This empirical observation is further supported by our theoretical analysis. In Appendix G, we prove that, in the absence of regularization, the trajectory score objective admits a degenerate constant-score solution as a global minimizer in the LfP and LfN settings. We further show that introducing the proposed regularization eliminates this degenerate optimum by ensuring that no constant score can globally minimize the objective. The empirical results demonstrate that the regularization is essential for learning reliable preference-weights.

6 Conclusion and Additional Experiments

We proposed PSIL, a unified offline safe imitation learning algorithm that leverages preferred, non-preferred, or both demonstrations. PSIL progressively propagates similarity information from these demonstrations to a larger trajectory dataset, which is then used in a weighted behavior cloning objective. Across the LfP, LfN, and LfP+N settings, our empirical results demonstrate that PSIL can learn from positive and/or negative signals to achieve favorable reward-safety trade-off. Additional experiments in Appendix F analyze the sensitivity to the weighting hyperparameters η and λ, as well as the number of non-preferred demonstrations in Appendix L.

AI use statement

In this work, we used generative AI tools for creating and editing software code, implementing methods, specifically for developing plotting functions. We have not used generative AI tools for ideation, developing theoretical models or conceptual frameworks, and formulating mathematical claims. Synthetic dataset generation is not applicable to this work. Additionally, we used generative

AI tools for editing the research paper to improve readability, and identifying relevant literature. We have reviewed all AI-assisted work. We manually reviewed and verified the LLM-generated codes and tested it for correctness. We also carefully reviewed all the AI-assisted paraphrasing to ensure that the meaning and technical content were preserved accurately. We used LLMs to identify and summarize relevant literature on Lipschitz continuity in transformer architectures and related work on safe imitation learning. All papers suggested by the LLMs were independently reviewed by the authors before being included in this work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Ethics statement

Our work has strong potential for real-world applications where safety is critical, such as autonomous driving and toxicity moderation in large language models. However, since our method relies on human-provided examples, it may also inherit human biases. Regulatory oversight may therefore be needed to help identify and mitigate these biases.

Reproducibility statement

Dataset related details are in Appendix A. Pseudo code, hyperparameter and computationally efficient implementation of our proposed algorithm, PSIL, are in Appendix B, C. Implementation details of IPL (LfP+N) and OSIL (LfP+N) are in Appendix D. Theoretical and empirical analysis of Lipschitz continuity of score function is in Appendix E. Theoretical and empirical analysis of regularization is in Appendix G.

References

Abbas Abdolmaleki, Bilal Piot, Bobak Shahriari, Jost Tobias Springenberg, Tim Hertweck, Michael Bloesch, Rishabh Joshi, Thomas Lampe, Junhyuk Oh, Nicolas Heess, Jonas Buchli, and Martin Riedmiller. Learning from negative feedback, or positive feedback or both. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4FVGowGzQb.

Eitan Altman. Constrained Markov Decision Processes. 1999. URL https://api.semanticscholar.org/CorpusID:14906227.

Massih-Reza Amini, Vasilii Feofanov, Loic Pauletto, Lies Hadjadj, Emilie Devijver, and Yury Maximov. Self-Training: A Survey. *Neurocomputing*, 616:128904, 2025.

Returaj Burnwal, Nirav Pravinbhai Bhatt, and Balaraman Ravindran. OSIL: Learning Offline Safe Imitation Policies with Safety Inferred from Non-preferred Trajectories. In *The 25th International Conference on Autonomous Agents and Multi-Agent Systems*, 2026a. URL https://openreview.net/forum?id=x2amyL6kNN.

Returaj Burnwal, Nirav Pravinbhai Bhatt, and Balaraman Ravindran. SafeMIL: Learning Offline Safe Imitation Policy from Non-Preferred Trajectories. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 19790–19798, 2026b.

Herbert Federer. *Geometric Measure Theory*. Classics in Mathematics. Springer Berlin Heidelberg, Berlin, Heidelberg, 1969. ISBN 9783642620102.

Justin Fu, Katie Luo, and Sergey Levine. Learning Robust Rewards with Adverserial Inverse Reinforcement Learning. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rkHywl-A-.

Divyansh Garg, Shuvam Chakraborty, Chris Cundy, Jiaming Song, and Stefano Ermon. IQ-Learn: Inverse Soft-Q Learning for Imitation. *Advances in Neural Information Processing Systems*, 34: 4028–4039, 2021.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Joey Hejna and Dorsa Sadigh. Inverse Preference Learning: Preference-Based RL Without a Reward Function. *Advances in Neural Information Processing Systems*, 36:18806–18827, 2023.

Jonathan Ho and Stefano Ermon. Generative Adversarial Imitation Learning. *Advances in Neural Information Processing Systems*, 29, 2016.

Huy Hoang, Tien Mai, and Pradeep Varakantham. Imitate the good and avoid the bad: An incremental approach to Safe Reinforcement Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 12439–12447, 2024.

Huy Hoang, Tien Anh Mai, and Pradeep Varakantham. No Experts, No Problem: Avoidance Learning from Bad Demonstrations. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=MYe8FiahWi.

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. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=toEGuA9Qfn.

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*, 2022. URL https://openreview.net/forum?id=BrPdX1bDZkQ.

Hyunjik Kim, George Papamakarios, and Andriy Mnih. The Lipschitz Constant of Self-Attention. In *International Conference on Machine Learning*, pp. 5562–5571. PMLR, 2021.

Ilya Kostrikov, Kumar Krishna Agrawal, Debidatta Dwibedi, Sergey Levine, and Jonathan Tompson. Discriminator-Actor-Critic: Addressing Sample Inefficiency and Reward Bias in Adversarial Imitation Learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Hk4fpoA5Km.

Ilya Kostrikov, Ofir Nachum, and Jonathan Tompson. Imitation Learning via Off-Policy Distribution Matching. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Hyg-JC4FDr.

Zuxin Liu, Zijian Guo, Yihang Yao, Zhepeng Cen, Wenhao Yu, Tingnan Zhang, and Ding Zhao. Constrained Decision Transformer for Offline Safe Reinforcement Learning. In *International Conference on Machine Learning*, pp. 21611–21630. PMLR, 2023.

Zuxin Liu, Zijian Guo, Haohong Lin, Yihang Yao, Jiacheng Zhu, Zhepeng Cen, Hanjiang Hu, Wenhao Yu, Tingnan Zhang, Jie Tan, and Ding Zhao. Datasets and Benchmarks for Offline Safe Reinforcement Learning. *Journal of Data-centric Machine Learning Research*, 2024.

Dean A Pomerleau. Efficient Training of Artificial Neural Networks for Autonomous Navigation. *Neural computation*, 3(1):88–97, 1991.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In *Proceedings of the fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

H. Scudder. Adaptive communication receivers. *IEEE Transactions on Information Theory*, 11(2): 167–174, 1965a. doi: 10.1109/TIT.1965.1053752.

H. Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965b. doi: 10.1109/TIT.1965.1053799.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-Training With Noisy Student Improves ImageNet Classification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10684–10695. IEEE, 2020.

Sheng Yue, Jiani Liu, Xingyuan Hua, Ju Ren, Sen Lin, Junshan Zhang, and Yaoxue Zhang. How to Leverage Diverse Demonstrations in Offline Imitation Learning. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Maryam Zare, Parham M Kebria, Abbas Khosravi, and Saeid Nahavandi. A Survey of Imitation Learning: Algorithms, Recent Developments, and Challenges. *IEEE Transactions on Cybernetics*, 54(12):7173–7186, 2024.

A Dataset Related Details

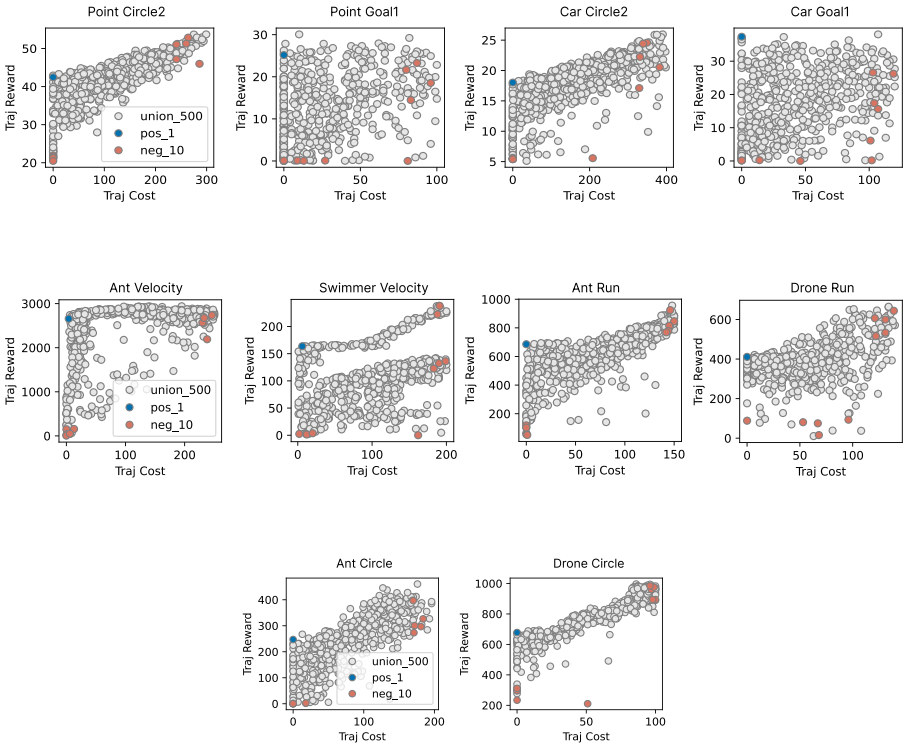

Figure 7: DSRL environments dataset. For each task, the figure shows one preferred trajectory in $\mathcal{D}_P$, ten non-preferred trajectories in $\mathcal{D}_N$, and $\mathcal{D}_U$ constructed from 500 trajectories randomly selected from the DSRL benchmark.

Table 1: Statistics of the preferred ($\mathcal{D}_P$), non-preferred ($\mathcal{D}_N$), and unlabeled ($\mathcal{D}_U$) trajectory datasets across the DSRL environments.

	Safety Gymnasium						Bullet-Gym			
Task	Circle2		Goal1		Velocity		Run		Circle	
Agent	Point	Car	Point	Car	Ant	Swimmer	Ant	Drone	Ant	Drone
Difficulty	2	2	1	1	-	-	-	-	-	-
Reward ($\mathcal{D}_P$)	42.47	18.01	25.15	37.35	2657.75	163.70	685.69	411.00	247.04	677.94
Cost ($\mathcal{D}_P$)	0.00	0.00	0.00	0.00	4.00	6.00	0.00	0.00	0.00	0.00
Reward ($\mathcal{D}_N$)	35.70	14.08	8.96	13.23	1419.64	67.54	445.31	269.13	181.54	594.79
Cost ($\mathcal{D}_N$)	130.30	222.30	55.90	66.00	121.10	119.80	73.30	95.10	100.10	57.40
Reward ($\mathcal{D}_U$)	41.51	17.39	12.39	17.69	2354.94	100.43	612.48	399.70	245.96	773.92
Cost ($\mathcal{D}_U$)	114.56	165.18	28.53	42.42	111.54	91.32	69.85	50.45	85.74	54.81

Table 2: Reward and cost bounds from the DSRL benchmark (`dsrl/infos.py`) used for normalization.

Suite	Task	$R_{\min}$	$R_{\max}$	$C_{\min}$	$C_{\max}$
Safety-Gymnasium	PointCircle2	20.48	54.02	0	299
	PointGoal1	0.01	30.07	0	100
	CarCircle2	5.11	25.95	0	398
	CarGoal1	0.01	39.91	0	120
	AntVelocity	6.15	2976.28	0	250
	SwimmerVelocity	0.07	238.96	0	200
Bullet-Gym	AntRun	0.0	955.48	0	150
	DroneRun	10.56	682.83	0	140
	AntCircle	0.02	460.71	0	200
	DroneCircle	207.79	996.39	0	100

To evaluate our approach, we conduct experiments on six Safety-Gymnasium tasks (POINT-CIRCLE2, CAR-CIRCLE2, POINT-GOAL1, CAR-GOAL1, ANT-VELOCITY, and SWIMMER-VELOCITY) and four Bullet-Gym tasks (DRONE-CIRCLE, DRONE-RUN, ANT-CIRCLE, and ANT-RUN), all from the official DSRL benchmark Liu et al. (2024). For each task, we construct $\mathcal{D}_P$ using one preferred trajectory, $\mathcal{D}_N$ using ten non-preferred trajectories, and $\mathcal{D}_U$ using 500 trajectories randomly selected from the DSRL benchmark, shown in Figure 7. Table 1 summarizes the mean cost, and reward for each domain used in our experiments. Table 2 report the reward and cost bounds from the DSRL benchmark.

B PSEUDOCODE AND IMPLEMENTATION DETAILS OF PSIL

Algorithm 1 PSIL

Require: Preferred ($\mathcal{D}_P$), Non-Preferred ($\mathcal{D}_N$), unlabeled ($\mathcal{D}_U$) trajectory dataset
Ensure: Learned Policy π

Initialize: score function s_θ, target score function $s_{\bar{\theta}}$, and policy π
Initialize: $w(\tau_P) = 1.0 \,\forall\, \tau_P \in \mathcal{D}_P$, $w(\tau_N) = -1.0 \,\forall\, \tau_N \in \mathcal{D}_N$, and $w(\tau_U) = 0.0 \,\forall\, \tau_U \in \mathcal{D}_U$
Initialize: $k = 0$
for $n = 1, 2, \ldots$ **do**
  Sample $\tau_P \sim \mathcal{D}_P$, $\tau_N \sim \mathcal{D}_N$, and $\tau_U \sim \mathcal{D}_U$
  if $k > 0$ **then**
    Update weight $w(\tau_U)$ using the target score function $s_{\bar{\theta}}$ according to Eq. 3 ..... (a)
  Optimize score function s_θ using Eq. 2
  Optimize policy π using Eq. 4
  if $n \bmod N = 0$, i.e., update every Nth-step **then** ..... (b)
    Update target score function: ..... (c)
      $\bar{\theta} \leftarrow \theta$
    $k \leftarrow k + 1$

Algorithm 1 provides the pseudocode for the practical implementation of PSIL. The procedure in the main paper describes trajectory weight assignment conceptually in terms of repeatedly optimizing the score function and propagating the resulting similarity information to the unlabeled dataset. In practice, we use two modifications to make this procedure computationally efficient.

First, rather than optimizing the score function until convergence at each iteration, we perform N gradient update steps before updating the preference-weights of the trajectory. Although convergence is useful for describing the iterative procedure, it is not necessary in practice. We found that optimizing the score function for a fixed number of steps provides a sufficient approximation while substantially reducing the computational cost (See Algo 1 (b)).

Second, directly updating the preference-weights of every trajectory in $\mathcal{D}_U$ after each score function training would require repeatedly evaluating the entire "unlabeled dataset". To avoid this additional

computational overhead, we maintain a target score function $s_{\bar{\theta}}$ that provides a temporally stable copy of the score function. During training, the weight of a sampled trajectory $\tau_U \in \mathcal{D}_U$ is updated online using target score function, $s_{\bar{\theta}}$, according to equation 3 (See Algo 1 (a)). Thus, the unlabeled dataset does not need to be revisited after every score function training. After every N steps, the target score function is updated toward the current score function (See Algo 1 (c)). The target score function therefore provides a delayed estimate, while the current score function continues to learn to the progressively expanded set of similar trajectories.

This implementation preserves the progressive nature of the proposed trajectory weight assignment: newly identified trajectories are assigned preference-weights based on the current similarity estimate, while previously assigned weights are retained (equation 3). The resulting procedure avoids full-dataset passes while retaining the iterative propagation of similarity information described in the main paper.

Hyperparameter. *Score function:* The score function is defined as $s_\theta(\tau) = \langle \mathbf{c}^\star, e_\theta(\tau) \rangle$. For $e_\theta : \tau \to \mathcal{S}^d$, we use a transformer architecture with two transformer blocks each with two attention heads. The model uses an embedding dimension (d) of 64 and a feed-forward dimension of 256 (4×64). We use causal self-attention, GELU activation, Layer Normalization, and residual connections. The final embedding representation is L2-normalized so that the representation lies on the unit hypersphere. For each random seed, we independently sample a non-learnable unit vector $\mathbf{c}^\star \in \mathcal{S}^d$ which remains fixed during training.

Policy Model: The policy is parameterized by a two-layer MLP (256 hidden units per layer) using ReLU activations.

Other hyperparameter: We use $\lambda = 0.99$, $\eta = 0.9$ and $\alpha = 0.2$ all our experiments, unless otherwise specified, such as in the ablation studies. Batch size 64, learning rate of 0.0003, $\gamma = 0.9$, L2-regularizer 0.01, and 1 million gradient steps.

C Variable-Length Trajectory and Policy Discounting

C.1 Handling Variable-Length Trajectories via a Sliding Window

We assume that trajectories have a fixed horizon H. However, environments in the DSRL benchmark generate variable-length episodes due to early termination upon task success or failure. To address this, we process the raw episodic trajectories using a sliding window approach. Given a unlabeled trajectory of length T and a target horizon H, we extract overlapping sub-trajectories of length H using a stride of 1. This generates exactly $T - H + 1$ sub-trajectories per episode. In our work, we use sub-trajectories of length $H = 50$.

C.2 Justification for Discounting in Policy Learning

In equation 4, the weighted behavior cloning objective applies a temporal discount factor γ^t to the state-action imitation loss. Since the score $s_\theta(\tau)$ is computed from the transformer output at the last state-action pair of the sub-trajectory, which attends to all preceding pairs, the preference-weight summarizes the entire sub-trajectory. Hence, the preference-weight of the sub-trajectory starting at (s_0, a_0) reflects the H steps of behavior that follow (s_0, a_0), similar to how a value function evaluates the future return from a state. For a later pair (s_t, a_t) in the sub-trajectory, the preference-weight reflects only part of its future. The temporal discount factor γ^t in equation 4 models this decaying relevance. Furthermore, since we extract sub-trajectories using a sliding window, every state-action pair from the original episode, except the last $H - 1$, occupies the $t = 0$ position for some sub-trajectory. Consequently, constraint violations or successful behaviors are penalized or reinforced when their corresponding sub-trajectory begins. The last $H - 1$ steps of each episode never appear at the beginning of a sub-trajectory and are therefore weighted only through earlier sub-trajectories.

D Implementation Details of IPL and OSIL for LfP+N setting

Both IPL (Hejna & Sadigh, 2023) and OSIL (Burnwal et al., 2026a) use preference-based objectives to infer task-relevant signals from trajectory comparisons. IPL leverages preference comparisons to learn a action-value function, whereas OSIL employs a Bradley-Terry objective to learn a cost model that assigns higher costs to non-preferred trajectories than to trajectories from the unlabeled dataset. In our setting, we construct additional preference pairs from the available trajectory sets. Specifically, since trajectories in $\mathcal{D}_U$ are expected to be safer on average than those in $\mathcal{D}_N$, we treat $\mathcal{D}_U$ as preferred to $\mathcal{D}_N$. When preferred demonstrations are available, we further impose the dataset-level ordering $\mathcal{D}_P \succ \mathcal{D}_U$ and $\mathcal{D}_P \succ \mathcal{D}_N$. These changes allow both IPL and OSIL to learn from both preferred and non-preferred demonstrations. We do not make any change in the policy learning procedure from the original algorithm. We used the authors implementation for both IPL and OSIL algorithm.

E Score Function Analysis

E.1 Choice of Score Function

E.1.1 Choice 1

Let $s_\theta(\tau) = \tanh\big(\text{MLP}_\theta(e_\theta(\tau))\big) \in [-1, 1]$, where $e_\theta : \tau \to \mathbb{R}^d$ is a learnable transformer-based network, and MLP_θ denotes a standard single-layer neural network. The $\tanh$ function is applied to ensure that the score function produces outputs bounded within $[-1, 1]$.

Standard self-attention is not globally Lipschitz continuous (Kim et al., 2021); however, it is Lipschitz continuous on bounded domain $\|\tau\| \leq l$, with a constant that grows with l. MLP is also Lipschitz continuous. In particular, there exist constants $K_1, K_2 > 0$ such that

$$\|e_\theta(\tau_1) - e_\theta(\tau_2)\|_2 \leq K_1\|\tau_1 - \tau_2\|, \quad \text{and} \quad |\text{MLP}_\theta(\mathrm{x}_1) - \text{MLP}_\theta(\mathrm{x}_2)| \leq K_2\|\mathrm{x}_1 - \mathrm{x}_2\|_2$$

where, $\mathrm{x} = e_\theta(\tau)$.

Furthermore, the $\tanh$ function is Lipschitz continuous with constant $K_{\tanh} = 1$. Therefore, by applying the following lemma on the composition of Lipschitz functions:

Lemma 1 (Federer (1969))**.** *Let g and h be two composable Lipschitz functions with Lipschitz constant $\text{Lip}(g)$ and $\text{Lip}(h)$. Then, $g \circ h$ is also Lipschitz continuous, with $\text{Lip}(g \circ h) \leq \text{Lip}(g)\,\text{Lip}(h)$.*

We can conclude that the composite function $s_\theta = \tanh \circ \text{MLP}_\theta \circ e_\theta$ is also Lipschitz continuous with constant bounded by $K = K_1K_2K_{\tanh}$ for bounded input trajectories. We refer this choice as "*No Projection*" based score function.

E.1.2 Choice 2

Let $\mathbf{c}^\star \in \mathcal{S}^d$ be a fixed random vector on the d-dimension unit hypersphere $\mathcal{S}^d$. We introduce a learnable transformer-based network $e_\theta : \tau \to \mathcal{S}^d$ and define the score function as $s_\theta(\tau) = \langle \mathbf{c}^\star, e_\theta(\tau)\rangle \in [-1, 1]$ which measures the cosine similarity between the frozen random unit vector and the learned trajectory embedding.

We can also show that dot product of fixed random unit vector $\mathbf{c}^\star$ and a transformer network is also Lipschitz continuous for bounded input.

Proposition 1. *Let $\mathbf{c}^\star$ be a fixed random unit vector and the transformer $e_\theta : \mathbb{R}^{H\times d} \to \mathbb{S}^d$ be Lipschitz continuous on the bounded domain $\|\tau\| \leq l$, with Lipschitz constants K. Define $s(\tau) = \langle \mathbf{c}^\star, e_\theta(\tau)\rangle$. Then,*

$$|s(\tau_1) - s(\tau_2)| \leq K\,\|\tau_1 - \tau_2\| \tag{9}$$

Proof.

$$\begin{aligned}
|s(\tau_1) - s(\tau_2)| &= |e_\theta(\tau_1)^\top \mathbf{c}^\star - e_\theta(\tau_2)^\top \mathbf{c}^\star| \\
&= |(e_\theta(\tau_1) - e_\theta(\tau_2))^\top \mathbf{c}^\star| \\
&\leq \|\mathbf{c}^\star\|_2 \, \|e_\theta(\tau_1) - e_\theta(\tau_2)\|_2 \\
&= \|e_\theta(\tau_1) - e_\theta(\tau_2)\|_2 \qquad \because \mathbf{c}^\star \in \mathbb{S}^d \text{ and } \|\mathbf{c}^\star\|_2 = 1 \\
&\leq K\|\tau_1 - \tau_2\| \qquad (10)
\end{aligned}$$

□

We refer this choice as "*Fixed Projection*" based score function.

Trajectory and Network Architecture. Each state-action pair dimension is normalized using statistics computed from $\mathcal{D}_U$, so that all trajectory features lie on a comparable scale and the Frobenius distance between trajectories is not dominated by any single feature.

While standard neural networks on bounded domains are theoretically Lipschitz continuous, unconstrained optimization can lead to arbitrarily large value of weight parameters, causing K to become large. To emperically constrain the magnitude of K, we use L2 regularization (weight decay) on the network weights. The Lipschitz constant of a linear neural network layer with weight matrix W is given by its spectral norm, which is defined as its largest singular value, $\sigma_{\max}$. L2 regularization actively penalizes the sum of the squared weights, which is the squared Frobenius norm, $\|W\|^2$. By definition, the Frobenius norm is the square root of the sum of all squared singular values and since singular values are non-negative, the largest singular value is strictly bounded:

$$\|W\| = \sqrt{\sum_i \sigma_i^2} \geq \sigma_{\max}$$

Therefore, penalizing the Frobenius norm through L2 regularization inherently penalizes an upper bound on the spectral norm (Goodfellow et al., 2016).

E.2 Implication of Lipschitz Property

Since score function is Lipschitz continuous, we can show why similar preferred or non-preferred trajectories are have similar value.

Corollary 1. *Let τ_1^p, τ_1^n be preferred and non-preferred trajectory sequences respectively and let τ_2^p and τ_2^n be some trajectory that satisfies $\|\tau_2^p - \tau_1^p\| \leq \epsilon$ and $\|\tau_2^n - \tau_1^n\| \leq \epsilon$. Then using Lipschitz continuity of the score function we can show*

$$|s(\tau_2^p) - s(\tau_1^p)| \leq K\epsilon \qquad \textit{and} \qquad |s(\tau_2^n) - s(\tau_1^n)| \leq K\epsilon$$

If ϵ is sufficiently small, then the score of τ_2^p will remain close to that of τ_1^p. An analogous statement holds for the non-preferred trajectory τ_1^n, τ_2^n. Therefore, ϵ close similar trajectories will be mapped closer to each other.

Limitation 1. *PSIL can fail due to following reasons: First, in complex environments, even a small perturbation can alter a trajectory's behavior sufficiently such that it is no longer preferred. Despite the change in behavior, the Lipschitz nature of the score function encourages ϵ-close trajectories to have similar value, as a result trajectories cannot be distinguished. Second, Property 1 does not guarantee the converse: trajectories that are far apart in trajectory space, and that may differ in behavior, can also receive similar scores. In both cases, the algorithm may incorrectly associate non-preferred trajectories with preferred ones (e.g., Car-Circle2, Figure 15).*

E.3 Comparison between different choices of score function

Table 3, 4 and Figure 8 compares the score functions based on *No Projection* (E.1.1) and *Fixed Projection* (E.1.2) in LfP+N setting. We observe that the *Fixed Projection-based* score function performs marginally better than the *No Projection-based* function. Moreover, the *No Projection-based* function introduces an additional learnable MLP layer, whereas the *Fixed Projection-based* function has no additional learnable parameters. Therefore, we use the *Fixed Projection-based* score function in all our experiments, as it achieves comparable performance with fewer parameters.

Table 3: **Reward and cost performance of the score function variants on Bullet Safety Gym in LfP+N setting.** We report the mean and standard deviation across five random seeds. Higher reward and lower cost are preferred.

Environment	Metric	PSIL (No Proj.)	**PSIL (Fixed Proj.)**
AntRun	Reward ↑	696.70 ± 19.96	684.01 ± 21.13
	Cost ↓	61.42 ± 9.85	55.36 ± 12.16
DroneRun	Reward ↑	369.64 ± 13.98	373.67 ± 16.78
	Cost ↓	36.79 ± 15.87	36.50 ± 10.94
AntCircle	Reward ↑	151.55 ± 8.51	203.47 ± 30.83
	Cost ↓	29.95 ± 10.39	37.18 ± 9.54
DroneCircle	Reward ↑	648.14 ± 9.99	654.55 ± 20.23
	Cost ↓	34.37 ± 3.95	39.04 ± 7.70

Table 4: **Reward and cost performance of the score function variants on Safety Gymnasium in LfP+N setting.** We report the mean and standard deviation across five random seeds. Higher reward and lower cost are preferred.

Environment	Metric	PSIL (No Proj.)	**PSIL (Fixed Proj.)**
PointCircle2	Reward ↑	38.13 ± 0.51	36.56 ± 1.59
	Cost ↓	33.13 ± 15.49	11.22 ± 6.24
PointGoal1	Reward ↑	10.94 ± 1.50	9.84 ± 0.94
	Cost ↓	13.72 ± 6.16	17.20 ± 4.63
CarCircle2	Reward ↑	19.44 ± 0.67	19.52 ± 0.84
	Cost ↓	206.10 ± 27.32	205.36 ± 26.38
CarGoal1	Reward ↑	13.08 ± 2.90	16.90 ± 1.95
	Cost ↓	22.63 ± 9.89	23.12 ± 11.64
AntVelocity	Reward ↑	2878.55 ± 14.18	2806.99 ± 90.65
	Cost ↓	10.12 ± 1.36	8.99 ± 1.43
SwimmerVelocity	Reward ↑	160.95 ± 0.67	157.83 ± 5.44
	Cost ↓	5.16 ± 3.00	6.38 ± 3.24

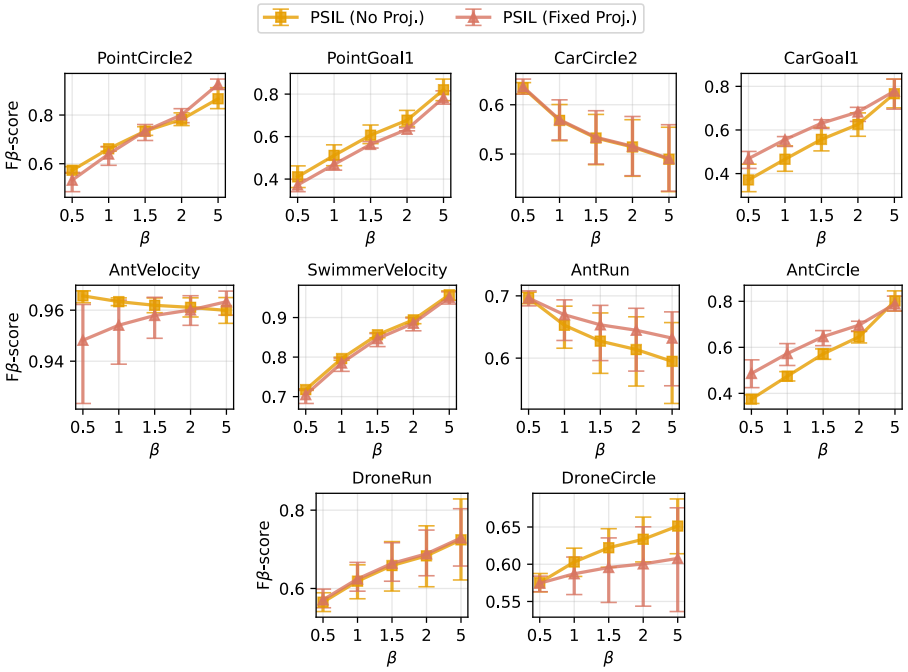

Figure 8: Comparison of the reward-safety trade-off across ten DSRL benchmark tasks in LfP+N setting. Each policy is trained for 1M gradient update steps and then evaluated using F_β-score with $\beta \in \{0.5, 1, 1.5, 2, 5\}$. Smaller values of β place greater emphasis on reward, whereas larger values emphasize safety. Fixed Projection based score function is slighly better than No Projection based score function. Error bars denote 95% bootstrap confidence intervals.

F SENSITIVITY ANALYSIS

F.1 SENSITIVITY ANALYSIS OF η

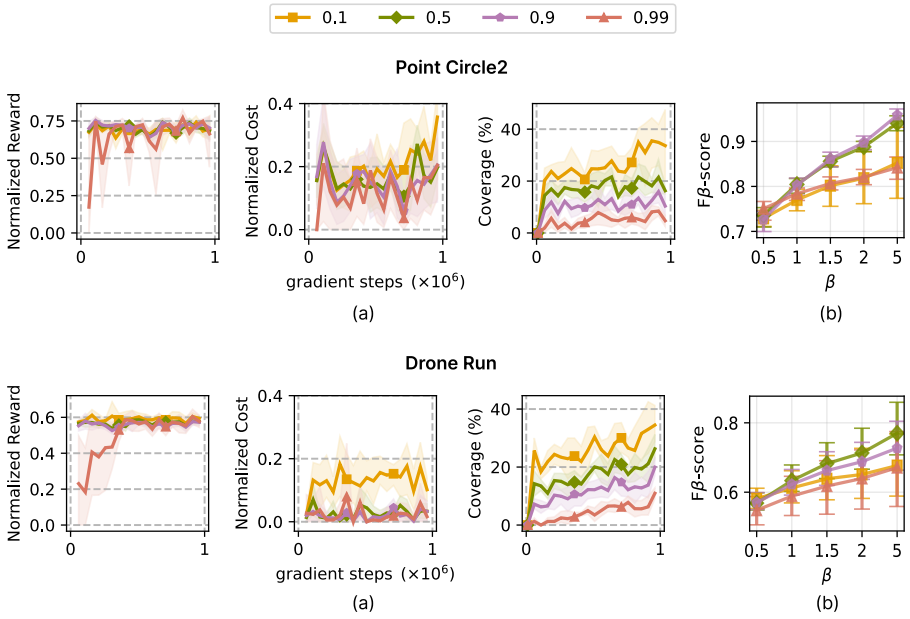

Figure 9: *Sensitivity to* η. (a) Normalized reward, cost and coverage of trajectories identified as preferred (i.e., assigned positive weights) for different values of η. (b) Corresponding $F\beta$-scores of the learned policies. Lower η increases trajectory coverage but admits more high-cost trajectories, while very large η identifies fewer trajectories and limits coverage. Error bars denote 95% bootstrap confidence intervals.

For our weighting procedure in equation 3, we introduce a hyperparameter $0 < \eta < 1$, which serves as a confidence threshold for determining whether a trajectory is sufficiently similar to receive a similarity pseudo-weight. To evaluate the sensitivity of our method to η, we vary its value and examine its effect on trajectory identification and final policy performance in the LfP+N setting, as shown in Figure 9. In Figure 9(a), we report the reward and cost only for the trajectories identified as preferred, i.e., the trajectories assigned positive weights, as these trajectories have a stronger influence on the weighted behavior cloning objective used for policy learning. Lower values of $\eta\,(= 0.1)$ increase trajectory coverage but also admit more high-cost trajectories into the preferred set, resulting in lower policy performance. In contrast, very large values of $\eta\,(= 0.99)$ yield a smaller set of high-quality trajectories, resulting in insufficient coverage for effective policy learning. An intermediate value, such as $\eta = 0.5$ or 0.9, provides a favorable balance between trajectory quality and coverage, thereby improving policy performance. These results indicate that high values of η generally identify high-quality trajectories for effective policy learning, but values too close to 1 require more training to identify a sufficient number of preferred trajectories and recover a comparable policy.

F.2 SENSITIVITY ANALYSIS OF λ

In our weighting procedure, equation 3, we introduce a hyperparameter $0 < \lambda < 1$, that controls the magnitude of the assigned pseudo-weights. To evaluate the sensitivity of our method to λ, we conduct experiments by varying its value and observing the effect on the final performance in LfP+N setting. Figure 10 shows the $F\beta$-score across different values of λ. The performance remains largely consistent across the evaluated values, indicating that PSIL is robust to the choice of λ when the corresponding scale of α is adjusted. This robustness is expected because λ primarily controls the scale of the preference-weights and does not change the relative ordering of trajectories. Thus, by appropriately adjusting α can compensate for changes in the scale induced by λ, leading to similar policy performance across different values of λ.

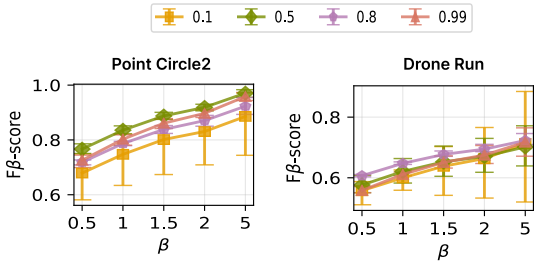

Figure 10: *Sensitivity to* λ. The $F\beta$-score remains largely consistent across different values of λ, indicating that PSIL is robust to the choice of λ when the corresponding scale of α is adjusted.

As discussed in the Hyperparameter section, we use $\lambda = 0.99$ and $\alpha = 0.2$ for all experiments as a default value. For the sensitivity analysis, we scale α proportionally with λ relative to this default value, i.e., $\alpha(\lambda) = 0.2\,\lambda/0.99$. This adjustment in α accounts for the change in preference-weight magnitude induced by λ. Therefore, the observed robustness suggests that λ primarily affects the preference-weight scale, whose effect on policy learning can be compensated by scaling the corresponding α.

G Regularization in Trajectory Weight Assignment

G.1 Theoretical Analysis of the Effect of Regularization

Let $s_\theta(\cdot) = \langle \mathbf{c}^\star, e_\theta(\cdot) \rangle$ be the score model, where $\mathbf{c}^\star$ is the fixed random unit vector and e_θ is our standard transformer network. Our goal is to learn θ such that the similar preferred trajectories are mapped to the $\mathbf{c}^\star$ direction and the similar non-preferred trajectories are mapped to the opposite direction. Therefore, the loss function is defined as follows.

$$\min_\theta \mathcal{L}_\theta := \sum_{\tau \in \mathcal{D}_P \cup \mathcal{D}_N \cup \mathcal{D}_U} \mathbb{1}_{w(\tau)>0} \left(1.0 - s_\theta(\tau)\right)^2 + \mathbb{1}_{w(\tau)<0} \left(-1.0 - s_\theta(\tau)\right)^2 + \underbrace{s_\theta(\tau_r)^2}_{\text{degeneracy regularizer}}$$

where τ_r is an element-wise convex combination of trajectories sampled from the unlabeled dataset.

Lemma 2. *Let $\mathcal{L}_\theta$ be defined as above.*

1. ***Without regularization.*** *When $\mathcal{D}_N = \emptyset$ and the regularization term is removed, the constant representation $e_\theta(\tau) = \mathbf{c}^\star,\ \forall \tau$, is a minimizer of the optimization problem.*
2. ***With regularization.*** *When $\mathcal{D}_N = \emptyset$ and the regularization term is included, any constant representation $e_\theta(\tau) = \mathbf{c},\ \forall \tau$, cannot be a global minimizer of $\mathcal{L}_\theta$.*

Proof. We first show that, in the absence of the regularization term, optimizing $\mathcal{L}_\theta$ produces a degenerate solution.

1. **Without regularization.**

   Suppose $e_\theta(\tau) = \mathbf{c}^\star,\ \forall \tau$. Then $s_\theta(\tau) = \langle e_\theta(\tau), \mathbf{c}^\star \rangle = 1$, for every trajectory τ. Consequently, $\mathcal{L}_\theta = \sum_{\mathcal{D}_P, \mathcal{D}_U} \mathbb{1}_{w(\tau)>0} \left(1.0 - s_\theta(\tau)\right)^2 = 0.0$. Therefore, the constant representation $e_\theta(\tau) = \mathbf{c}^\star$ is a valid minimizer. In this case, the score function loses all discriminative power since every trajectory is mapped to the same representation.

2. **With regularization.**

   We prove by contradiction. Suppose there exists a constant representation $e_\theta(\tau) = \mathbf{c},\ \forall \tau$, that globally minimizes $\mathcal{L}_\theta$. Consider the following cases.

   Case 1: $\mathbf{c} = \mathbf{c}^\star$ and $\mathcal{D}_N = \emptyset$. Then all random trajectories satisfy $s_\theta(\tau_r) = 1$, and therefore

   $$\mathcal{L}_\theta = \sum_{\mathcal{D}_P, \mathcal{D}_U} \left(s_\theta(\tau_r)\right)^2 > 0$$

.

Case 2: $\mathbf{c} \neq \mathbf{c}^\star$ and $\mathcal{D}_N = \emptyset$. In this case, $s_\theta(\tau) = \mathbf{c}^\top \mathbf{c}^\star,\ \forall \tau$, and hence

$$\mathcal{L}_\theta = \sum_{\mathcal{D}_P, \mathcal{D}_U} \mathbb{1}_{w(\tau)>0} \, (1.0 - \mathbf{c}^\top \mathbf{c}^\star)^2 + (\mathbf{c}^\top \mathbf{c}^\star)^2 > 0$$

Thus, in both cases the loss remains strictly positive.

Case 3: To establish that a constant representation cannot be a global minimizer, it suffices to show that a non-constant representation can achieve a strictly smaller loss. Consider an embedding function satisfying $e_\theta(\tau) = \mathbf{c}^\star$ for $w(\tau) > 0$ and $e_\theta(\tau_r) = \mathbf{c}^\perp$ for convex combination trajectories, where $\mathbf{c}^\perp$ is a unit vector orthogonal to $\mathbf{c}^\star$, i.e., $(\mathbf{c}^\perp)^\top \mathbf{c}^\star = 0$. Then $s_\theta(\tau) = \langle \mathbf{c}^\star, e_\theta(\tau) \rangle = 1$ for $w(\tau) > 0$ and $s_\theta(\tau_r) = \langle \mathbf{c}^\star, e_\theta(\tau_r) \rangle = 0$. Consequently, every term in the objective is equal to zero, implying $\mathcal{L}_\theta = 0$. Since any constant representation incurs a strictly positive loss while the above non-constant representation achieves zero loss, no constant representation can be a global minimizer of $\mathcal{L}_\theta$.

□

A symmetric argument holds for the LfN setting ($\mathcal{D}_P = \emptyset$), with $\mathbf{c}^\star$ replaced by $-\mathbf{c}^\star$. Without regularization, the constant representation $e_\theta(\tau) = -\mathbf{c}^\star$, which gives $s_\theta(\tau) = -1$ for all τ, is a global minimizer. With regularization, no constant representation can be a global minimizer.

Choice of regularizer. Let $\mathbb{M} = \left\{ \tau = (s_0, a_0, s_1, a_1, \ldots) : s_{t+1} \sim \mathcal{T}(s_t, a_t) \ \forall t \right\}$ denote the set of trajectories consistent with the environment dynamics. Because $\mathbb{M}$ is generally a non-convex set, an element-wise convex combination of state-action pairs from two trajectories $\tau_1, \tau_2 \in \mathbb{M}$ is not guaranteed to lie in $\mathbb{M}$.

We exploit this property of $\mathbb{M}$ by sampling τ_1, τ_2 independently from the unlabeled dataset $\mathcal{D}_U$, and forming

$$\tau_r = \boldsymbol{\zeta} \odot \tau_1 + (\mathbf{1} - \boldsymbol{\zeta}) \odot \tau_2$$

where $\boldsymbol{\zeta} \sim U(0,1)^{|\tau|}$ is a tensor of independent uniform random variables matching the shape of the trajectory, $\mathbf{1}$ is a tensor of ones, and $\odot$ denotes the Hadamard (element-wise) product. Applying element-wise interpolation disrupts the temporal and internal geometric structure of the state-action pairs, resulting in a τ_r that generally violates the dynamics constraints, meaning $\tau_r \notin \mathbb{M}$. Such a τ_r is not a trajectory that the dynamics model can produce and consequently has no well-defined label: it is neither meaningfully preferred nor non-preferred. This motivates the regularizer $s_\theta(\tau_r)^2$, which softly enforces $s_\theta(\tau_r) \approx 0.0$. The penalty anchors the score function by suppressing the scores of generally invalid trajectories rather than allowing them to assign an arbitrarily strong positive or negative score.

Limitation. While penalizing these unrealizable trajectories, τ_r, successfully prevents degenerate solutions, the learned s_θ solution using this specific heuristic may still be suboptimal. Designing a theoretically optimal regularization scheme remains an open direction for future work.

G.2 Additional Empirical Evidence

G.2.1 Effect of Regularization in LfP Setting

We additionally evaluate the effect of regularization on the DroneRun environment in Figure 11. Consistent with the PointCircle2 results in the main paper, removing the regularization term causes the trajectory score model to collapse toward a nearly constant prediction close to 1.0. As a consequence, the resulting weights fail to distinguish between trajectories of different quality and assign large positive weights even to high-cost trajectories. With regularization, the score model remains discriminative, leading to more reliable trajectory weight assignment and safer policy learning. These results further support the importance of the proposed regularization for avoiding score collapse in the LfP setting.

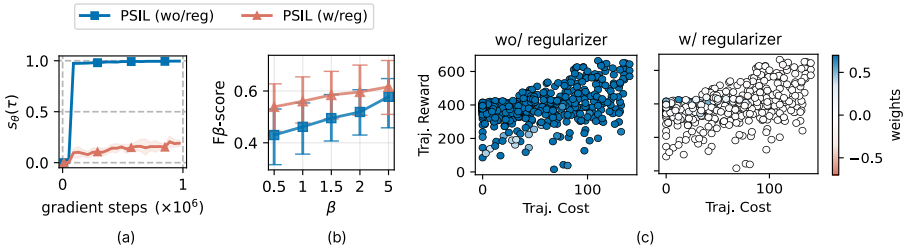

Figure 11: *Effect of regularization in the LfP setting for DroneRun environment.* (a) Trajectory score $s_\theta(\tau)$ during training with and without regularization. Without regularization, the score model collapses to a nearly constant prediction close to 1.0 for all trajectories. (b) Removing the regularization term results in lower $F\beta$-scores. (c) Final trajectory weights after training. Without regularization, the collapsed score model assigns large positive weights to even high-cost trajectories. In contrast, the proposed regularization learns discriminative trajectory scores, resulting in accurate trajectory weight assignment and effective separation of preferred and non-preferred behaviors. Error bars denote 95% bootstrap confidence interval.

G.2.2 Effect of Regularization in LfN Setting

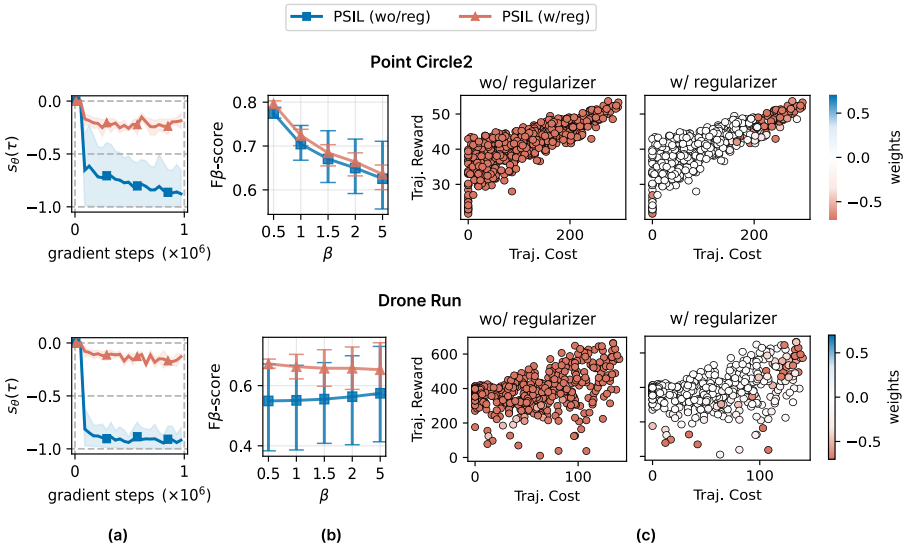

Figure 12: *Effect of regularization in the LfN setting for PointCircle2 and DroneRun environment.* (a) Trajectory score $s_\theta(\tau)$ during training with and without regularization. Without regularization, the score model collapses to a nearly constant prediction close to -1.0 for all trajectories. (b) Removing the regularization term results in lower $F\beta$-scores. (c) Final trajectory weights after training. Without regularization, the collapsed score model assigns large negative weights to even low-cost trajectories. In contrast, the proposed regularization learns discriminative trajectory scores, resulting in accurate trajectory weight assignment and effective separation of preferred and non-preferred behaviors. Error bars denote 95% bootstrap confidence interval.

We evaluate the effect of the proposed regularization in the LfN setting, where non-preferred demonstrations $\mathcal{D}_N$ are available while $\mathcal{D}_P = \emptyset$. Results on PointCircle2, DroneRun are in Figure 12.

Similar to the LfP setting, removing the regularization term leads to a degenerate solution. However, because supervision in the LfN setting is provided exclusively by non-preferred demonstrations, the score model collapses toward a nearly constant prediction close to -1.0. Consequently, the resulting trajectory weights become poorly discriminative, assigning large negative weights even to trajectories with relatively low cost and desirable task performance. This prevents the score model from reliably distinguishing trajectories according to its similarity to the non-preferred demonstrations.

In contrast, incorporating the proposed regularization prevents this score collapse and preserves meaningful variation in the trajectory scores. The resulting trajectory weights provide a more informative ordering of the offline trajectories and lead to improved policy performance as measured by Fβ. The consistent behavior observed across both PointCircle2 and DroneRun further demonstrates that the role of the regularization is not specific to a particular environment, but is important for avoiding degenerate score solutions under one-sided supervision.

G.2.3 Effect of Regularization in LfP+N Setting

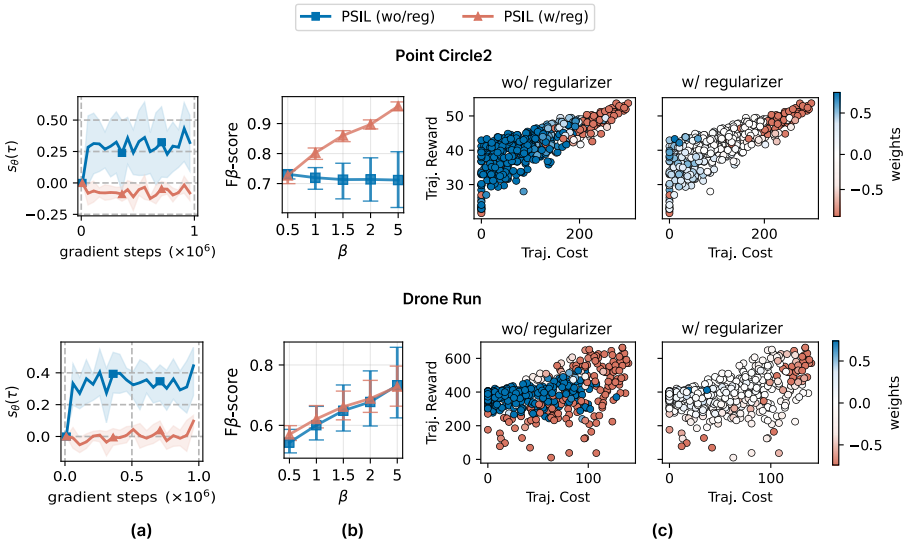

Figure 13: *Effect of regularization in the LfP+N setting on PointCircle2 and DroneRun.* (a) Trajectory scores $s_\theta(\tau)$ during training with and without regularization. In the LfP+N setting, the score model does not collapse to a constant solution even without regularization, as indicated by the variation in trajectory scores. Regularization results in lower score values due to the additional penalty on the score of the convex-combination trajectory τ_r. (b) Removing regularization leads to lower $F\beta$-scores. (c) Final trajectory weights after training. Without regularization, intermediate trajectories receive relatively large-magnitude weights, despite having lower similarity confidence to the preferred and non-preferred demonstrations. Regularization reduces the magnitude of these weights, resulting in better trajectory discrimination. Error bars denote 95% bootstrap confidence intervals.

We investigate the effect of the regularization term in the LfP+N setting, where both preferred and non-preferred demonstrations are available. Figure 13 compares the score model with and without regularization on PointCircle2 and DroneRun. Unlike the LfP and LfN settings, the score model does not exhibit a degenerate constant solution in the absence of regularization. As shown in Figure 13(a), the trajectory scores retain substantial variance throughout training. This suggests that the presence of both preferred and non-preferred demonstrations provides sufficient supervision to avoid score collapse in the LfP+N setting.

Nevertheless, regularization still affects the resulting trajectory weights and policy performance. With regularization, the score values are generally lower due to the additional penalty applied to the score of the convex-combination trajectory τ_r, equation 2. More importantly, without regularization, intermediate trajectories receive relatively large-magnitude weights despite having lower similarity confidence to the preferred and non-preferred demonstrations. This can cause these trajectories to have an overly strong influence during policy learning, resulting in lower $F\beta$-scores, as shown in Figure 13(b). In contrast, regularization reduces the magnitude of the weights assigned to such intermediate trajectories (Figure 13(c)), leading to better trajectory discrimination and improved policy performance.

Overall, these results indicate that, in the LfP+N setting, regularization is not necessary to prevent score collapse, but remains beneficial for controlling the magnitude of propagated trajectory weights and their influence during policy learning.

H Additional Empirical Evidence for PSIL Trajectory Weight Assignment

We analyze the training behavior of PSIL in the DSRL benchmarks. Figures 14 and 15 show that PSIL progressively identifies trajectories from the unlabeled dataset $\mathcal{D}_U$ that are similar to the preferred and non-preferred demonstrations. In general, the identified preferred trajectories exhibit high reward and low cost, whereas the identified non-preferred trajectories are low reward or high cost.

Furthermore, Figures 14(b) and 15(b) show the final propagated similarity weights across the unlabeled dataset. The learned weights are consistent: trajectories with high reward and low cost receive large positive weights, while trajectories with low reward or high cost receive large negative weights. These results support the ability of the progressive weighting procedure to propagate the similarity information from a small number of demonstrations to a substantially larger set of trajectories.

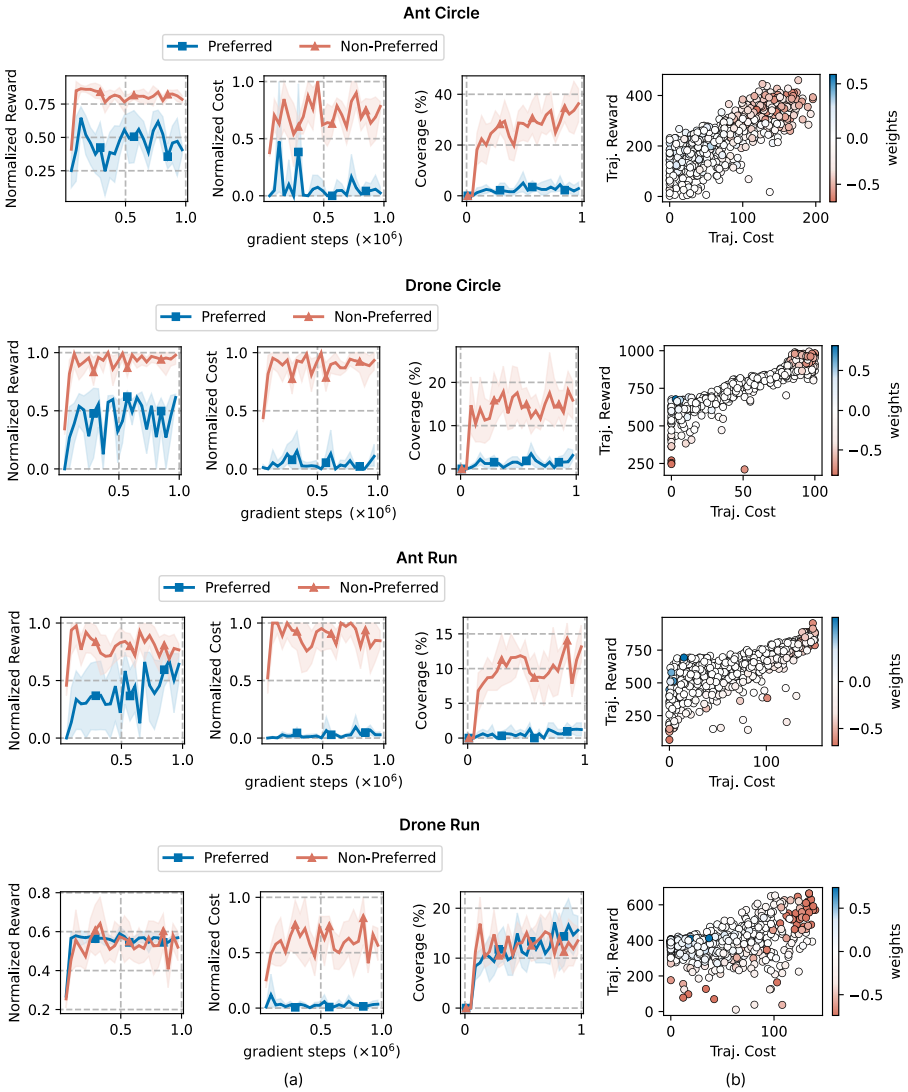

Figure 14: *Progressive Trajectory Weight Assignment for Bullet Safety Gym.* (a) As training progresses, PSIL identifies trajectories in $\mathcal{D}_U$ similar to the preferred and non-preferred demonstrations. (b) Final trajectory weights after training. Preferred trajectories with high reward and low cost receive large positive weights, while non-preferred trajectories with low reward or high cost receive large negative weights.

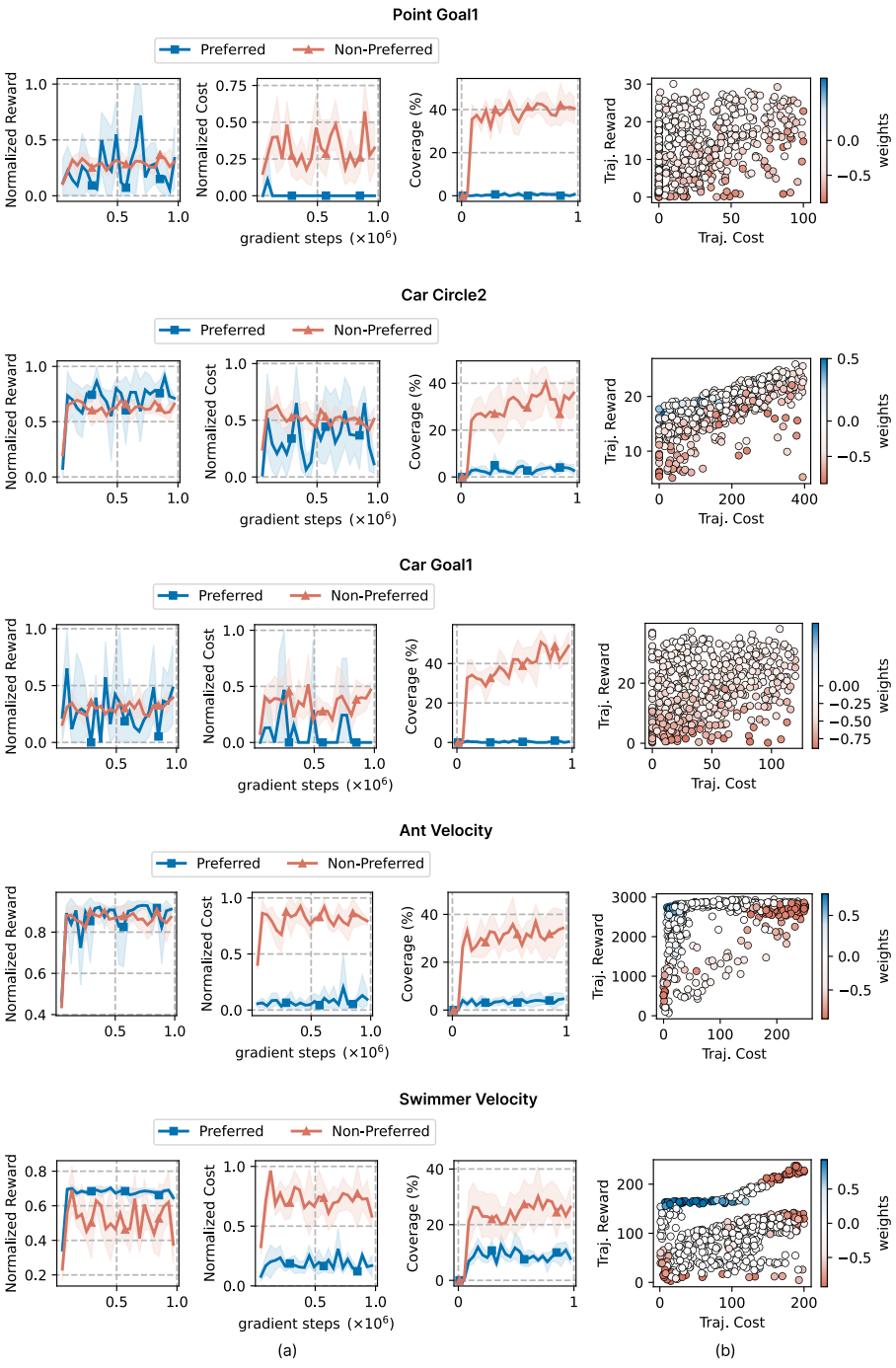

Figure 15: *Progressive Trajectory Weight Assignment for Safety Gymnasium Environment.* (a) As training progresses, PSIL identifies trajectories in $\mathcal{D}_U$ similar to the preferred and non-preferred demonstrations. (b) Final trajectory weights after training. Preferred trajectories with high reward and low cost receive large positive weights, while non-preferred trajectories with low reward or high cost receive large negative weights.

I ADDITIONAL RESULTS ON LFP+N SETTING

Table 5: **Reward and cost performance on Safety Gymnasium (LfP+N).** We report the mean and standard deviation across five random seeds. Higher reward and lower cost are preferred. These raw reward and cost results complement the Fβ results reported in the main paper, which directly evaluate the reward–safety trade-off.

Environment	Metric	SafeDICE	UNIQ	OSIL (LfP+N)	IPL (LfP+N)	**PSIL (LfP+N)**
PointCircle2	Reward ↑	46.79 ± 0.81	22.01 ± 3.75	41.18 ± 0.70	43.57 ± 1.17	36.56 ± 1.59
	Cost ↓	221.77 ± 9.93	8.29 ± 9.73	88.43 ± 13.65	133.77 ± 12.41	11.22 ± 6.24
PointGoal1	Reward ↑	18.68 ± 1.79	9.12 ± 1.95	14.32 ± 2.48	10.20 ± 1.91	9.84 ± 0.94
	Cost ↓	28.85 ± 5.75	20.51 ± 8.57	29.91 ± 9.46	40.84 ± 6.38	17.20 ± 4.63
CarCircle2	Reward ↑	20.28 ± 0.73	7.61 ± 0.72	19.25 ± 0.97	19.65 ± 0.87	19.52 ± 0.84
	Cost ↓	241.40 ± 37.46	30.29 ± 19.51	221.62 ± 23.64	211.19 ± 38.49	205.36 ± 26.38
CarGoal1	Reward ↑	14.93 ± 2.96	10.31 ± 1.61	15.37 ± 2.46	15.81 ± 3.92	16.90 ± 1.95
	Cost ↓	25.64 ± 9.34	22.84 ± 9.28	21.28 ± 1.64	30.97 ± 11.03	23.12 ± 11.64
AntVelocity	Reward ↑	2893.72 ± 71.59	2761.10 ± 31.60	2869.52 ± 12.60	2872.78 ± 73.96	2806.99 ± 90.65
	Cost ↓	55.21 ± 10.63	4.23 ± 2.77	21.31 ± 4.85	84.49 ± 10.31	8.99 ± 1.43
SwimmerVelocity	Reward ↑	115.62 ± 51.64	25.38 ± 8.39	66.80 ± 45.99	68.01 ± 23.72	157.83 ± 5.44
	Cost ↓	200.00 ± 107.78	141.87 ± 64.61	103.61 ± 37.83	19.50 ± 6.32	6.38 ± 3.24

Table 6: **Reward and cost performance on Bullet Safety Gym (LfP+N).** We report the mean and standard deviation across five random seeds. Higher reward and lower cost are preferred. These raw reward and cost results complement the Fβ results reported in the main paper, which directly evaluate the reward–safety trade-off.

Environment	Metric	SafeDICE	UNIQ	OSIL (LfP+N)	IPL (LfP+N)	**PSIL (LfP+N)**
AntRun	Reward ↑	732.75 ± 10.44	269.83 ± 20.66	71.24 ± 16.07	705.16 ± 15.33	684.01 ± 21.13
	Cost ↓	75.70 ± 6.50	7.03 ± 1.45	0.08 ± 0.16	60.17 ± 12.53	55.36 ± 12.16
DroneRun	Reward ↑	395.60 ± 6.72	383.52 ± 59.86	237.26 ± 186.84	417.44 ± 36.83	373.67 ± 16.78
	Cost ↓	9.39 ± 8.97	108.89 ± 9.89	53.31 ± 45.13	66.22 ± 29.94	36.50 ± 10.94
AntCircle	Reward ↑	238.42 ± 23.04	82.22 ± 9.69	160.65 ± 112.94	244.64 ± 30.48	203.47 ± 30.83
	Cost ↓	58.61 ± 11.31	16.51 ± 5.93	42.71 ± 31.03	64.21 ± 17.01	37.18 ± 9.54
DroneCircle	Reward ↑	719.64 ± 11.92	77.10 ± 31.75	746.49 ± 21.95	744.38 ± 24.16	654.55 ± 20.23
	Cost ↓	37.88 ± 2.32	12.33 ± 8.39	42.52 ± 4.98	49.12 ± 3.10	39.04 ± 7.70

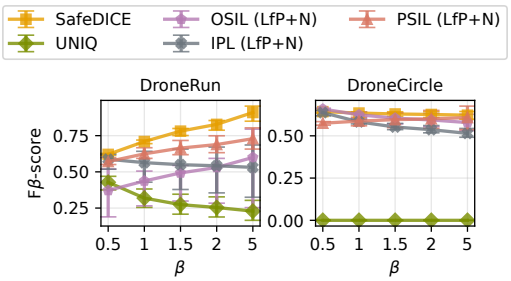

Figure 16: Comparison of the reward-safety trade-off on DRONE CIRCLE and DRONE RUN task. Each policy is trained for 1M gradient update steps and then evaluated using Fβ-score with $\beta \in \{0.5, 1, 1.5, 2, 5\}$. Smaller values of β place greater emphasis on reward, whereas larger values emphasize safety. Methods that maintain high Fβ-scores across different β values indicate a better reward-safety trade-off. Error bars denote 95% bootstrap confidence intervals.

Figure 16 reports the Fβ score for AntCircle and DroneCircle tasks. Tables 5 and 6 report the raw total reward and cost of the learned policies on Safety Gymnasium and Bullet Safety Gym, respectively. We report the mean and standard deviation across five random seeds. Higher reward and lower cost are preferred.

These raw metrics provide complementary information about policy performance, but should not be interpreted independently. In particular, a policy can achieve a low cost by adopting overly conservative behavior that substantially reduces task performance. Conversely, a policy can achieve high reward while incurring a large safety cost. Thus, comparing methods using reward or cost alone can favor policies that optimize one objective at the expense of the other. For example, on several environments, PSIL achieves a relatively high reward while maintaining substantially lower cost than methods that obtain comparable task performance. At the same time, some competing methods achieve even lower cost by sacrificing reward. The raw reward and cost values therefore do not necessarily yield a consistent ranking of the methods.

Across the evaluated environments, PSIL generally provides a favorable balance between reward and safety. Its advantage is therefore better characterized as a reward-safety trade-off rather than dominance in either raw reward or raw cost. The corresponding Fβ results in the main paper capture this trade-off directly and provide the primary basis for comparing the overall quality of the learned policies. Additionally, we also report the learning curves for PSIL algorithm in Figure 17, 18, 19.

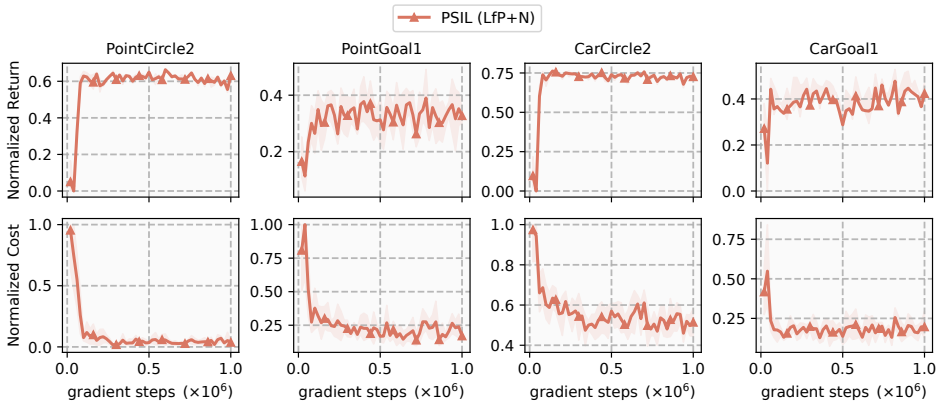

Figure 17: *Learning Curve of PSIL algorithm (LfP+N).* Experimental results on POINT CIRCLE2, POINT GOAL1, CAR CIRCLE2, and CAR GOAL1 tasks. The shaded area represents the standard error.

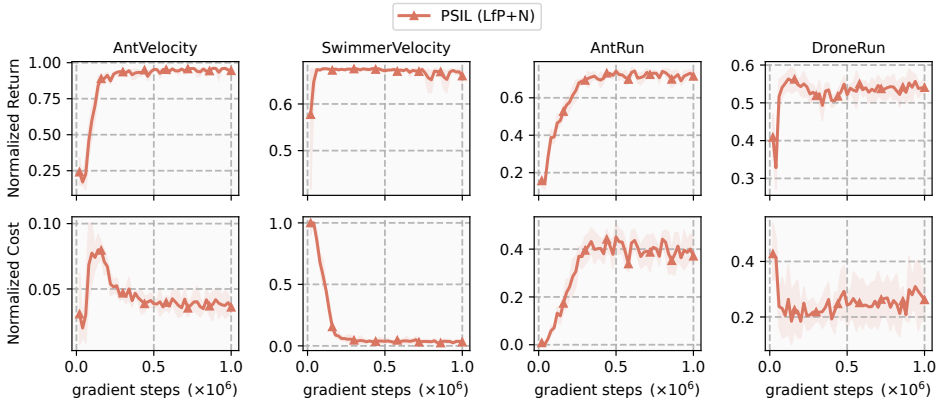

Figure 18: *Learning Curve of PSIL algorithm (LfP+N).* Experimental results on ANT VELOCITY, SWIMMER VELOCITY, ANT RUN, and DRONE RUN tasks. The shaded area represents the standard error.

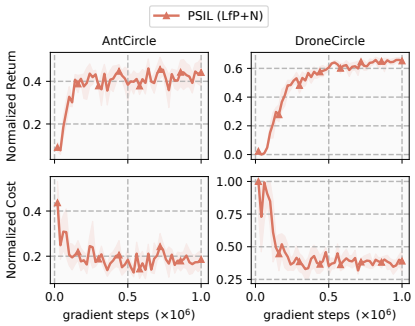

Figure 19: *Learning Curve of PSIL algorithm (LfP+N).* Experimental results on ANT CIRCLE, DRONE CIRCLE tasks. The shaded area represents the standard error.

J ADDITIONAL RESULTS ON LFP SETTING

Table 7: **Reward and cost performance on Safety Gymnasium (LfP).** We report the mean and standard deviation across five random seeds. Higher reward and lower cost are preferred. These raw reward and cost results complement the Fβ results reported in the main paper, which directly evaluate the reward–safety trade-off.

Environment	Metric	BC-Comp	IQ-Learn	ILID	IPL (LfP)	**PSIL (LfP)**
PointCircle2	Reward ↑	19.11 ± 0.81	42.14 ± 5.02	35.32 ± 2.88	41.15 ± 0.67	39.98 ± 1.20
	Cost ↓	1.19 ± 1.89	172.33 ± 77.95	92.66 ± 51.02	52.98 ± 26.38	79.34 ± 43.72
PointGoal1	Reward ↑	6.22 ± 1.59	9.81 ± 1.30	4.69 ± 2.23	11.87 ± 0.82	14.04 ± 1.66
	Cost ↓	18.98 ± 4.58	22.11 ± 2.13	51.69 ± 18.77	22.76 ± 5.06	20.55 ± 6.29
CarCircle2	Reward ↑	6.39 ± 0.46	15.51 ± 1.54	5.36 ± 1.44	18.84 ± 0.77	19.62 ± 0.70
	Cost ↓	11.89 ± 4.98	245.43 ± 37.61	367.10 ± 25.70	180.22 ± 30.20	205.13 ± 22.55
CarGoal1	Reward ↑	9.02 ± 1.71	9.94 ± 4.15	21.43 ± 0.92	15.74 ± 1.24	15.18 ± 3.53
	Cost ↓	40.56 ± 20.28	18.46 ± 1.90	63.77 ± 11.36	17.36 ± 7.50	18.34 ± 7.12
AntVelocity	Reward ↑	2000.08 ± 140.78	1172.96 ± 119.97	2753.24 ± 122.15	2795.77 ± 109.55	2820.40 ± 80.82
	Cost ↓	56.62 ± 7.50	20.29 ± 5.27	9.01 ± 3.19	10.51 ± 1.49	23.26 ± 12.50
SwimmerVelocity	Reward ↑	46.63 ± 7.53	100.41 ± 27.44	147.21 ± 8.72	72.51 ± 35.40	160.99 ± 0.31
	Cost ↓	137.67 ± 9.71	148.22 ± 84.44	38.37 ± 22.99	53.59 ± 49.61	7.31 ± 3.37

Table 8: **Reward and cost performance on Bullet Safety Gym (LfP).** We report the mean and standard deviation across five random seeds. Higher reward and lower cost are preferred. These raw reward and cost results complement the Fβ results reported in the main paper, which directly evaluate the reward–safety trade-off.

Environment	Metric	BC-Comp	IQ-Learn	ILID	IPL (LfP)	**PSIL (LfP)**
AntRun	Reward ↑	252.03 ± 35.53	140.15 ± 17.47	148.03 ± 20.66	650.82 ± 57.00	701.18 ± 28.85
	Cost ↓	8.57 ± 3.95	0.32 ± 0.46	1.44 ± 0.94	44.91 ± 18.93	67.75 ± 9.90
DroneRun	Reward ↑	407.71 ± 78.54	379.74 ± 80.07	364.77 ± 62.83	315.57 ± 154.30	365.94 ± 73.42
	Cost ↓	99.54 ± 4.28	83.88 ± 22.52	76.57 ± 16.73	55.99 ± 24.73	52.38 ± 16.38
AntCircle	Reward ↑	52.26 ± 5.15	171.79 ± 23.57	86.03 ± 14.42	211.42 ± 25.16	205.35 ± 24.56
	Cost ↓	33.38 ± 10.94	78.25 ± 15.81	50.13 ± 7.15	44.42 ± 8.72	53.43 ± 12.23
DroneCircle	Reward ↑	119.14 ± 35.60	53.47 ± 43.81	174.47 ± 103.95	679.80 ± 19.64	668.19 ± 57.77
	Cost ↓	22.55 ± 2.35	56.82 ± 15.38	21.09 ± 15.21	32.53 ± 5.35	44.98 ± 5.62

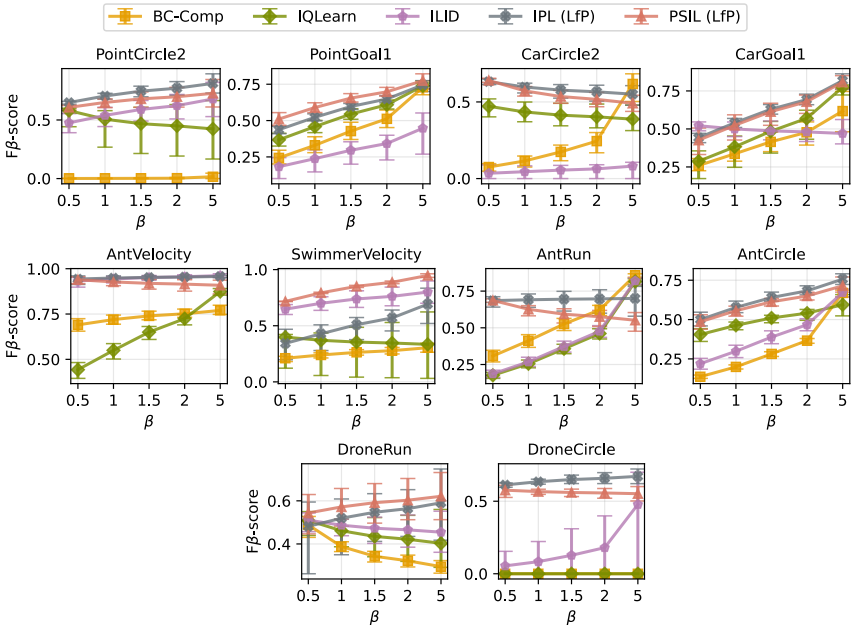

Figure 20: Comparison of the reward-safety trade-off across ten DSRL benchmark tasks in LfP setting. Each policy is trained for 1M gradient update steps and then evaluated using Fβ-score with $\beta \in \{0.5, 1, 1.5, 2, 5\}$. Smaller values of β place greater emphasis on reward, whereas larger values emphasize safety. Methods that maintain high $F\beta$-scores across different β values indicate a better reward-safety trade-off. Error bars denote 95% bootstrap confidence intervals.

Tables 7 and 8 report the raw total reward and cost of the learned policies on Safety Gymnasium and Bullet Safety Gym, respectively in LfP setting. In Figure 20, we report the Fβ scores for the DSRL benchmark environments. Using only preferred demonstrations, PSIL obtains reward-cost trade-offs competitive with the strongest baseline, IPL, although IPL achieves higher reward and lower cost on PointCircle2, AntCircle, and DroneCircle. By progressively assigning higher similarity weights to trajectories that resemble the preferred demonstrations, PSIL is able to exploit the entire offline dataset to recover safe policies using only a small set of preferred demonstrations.

K Additional Results on LfN Setting

Tables 9 and 10 report the raw reward-return and cost-return values of the learned policies on the Safety Gymnasium and Bullet Safety Gym environments, respectively, and Figure 21 reports the per-task Fβ-scores. In the LfN setting, PSIL achieves performance comparable to existing LfN methods. Since only non-preferred supervision is available, PSIL readily identifies and down-weights trajectories that are highly similar to the non-preferred demonstrations. However, intermediate trajectories, which are less similar to $\mathcal{D}_N$, remain unlabeled and retain non-negligible influence on the weighted behavior cloning objective (Figure 12(c)). Because the preferred behavior lies furthest from the non-preferred demonstrations, it is identified only after negative weights have propagated through most of $\mathcal{D}_U$. This limits the safety gains achievable from non-preferred demonstrations alone.

Table 9: **Reward and cost performance on Safety Gymnasium (LfN).** We report the mean and standard deviation across five random seeds. Higher reward and lower cost are preferred. These raw reward and cost results complement the Fβ results reported in the main paper, which directly evaluate the reward–safety trade-off.

Environment	Metric	SafeDICE	UNIQ	OSIL (LfN)	**PSIL (LfN)**
PointCircle2	Reward ↑	46.79 ± 0.81	22.01 ± 3.75	42.95 ± 1.10	43.89 ± 0.21
	Cost ↓	221.77 ± 9.93	8.29 ± 9.73	146.22 ± 20.92	157.95 ± 15.03
PointGoal1	Reward ↑	18.68 ± 1.79	9.12 ± 1.95	14.92 ± 1.06	10.66 ± 1.78
	Cost ↓	28.85 ± 5.75	20.51 ± 8.57	23.09 ± 5.67	15.90 ± 5.92
CarCircle2	Reward ↑	20.28 ± 0.73	7.61 ± 0.72	20.27 ± 0.72	20.03 ± 1.19
	Cost ↓	241.40 ± 37.46	30.29 ± 19.51	229.65 ± 32.11	236.77 ± 29.57
CarGoal1	Reward ↑	14.93 ± 2.96	10.31 ± 1.61	14.71 ± 4.28	15.57 ± 2.84
	Cost ↓	25.64 ± 9.34	22.84 ± 9.28	16.55 ± 6.59	19.13 ± 9.45
AntVelocity	Reward ↑	2893.72 ± 71.59	2761.10 ± 31.60	2897.00 ± 48.33	2949.48 ± 14.33
	Cost ↓	55.21 ± 10.63	4.23 ± 2.77	21.05 ± 15.23	65.67 ± 19.13
SwimmerVelocity	Reward ↑	115.62 ± 51.64	25.38 ± 8.39	114.04 ± 17.41	130.75 ± 10.81
	Cost ↓	200.00 ± 107.78	141.87 ± 64.61	153.06 ± 74.74	63.02 ± 19.84

Table 10: **Reward and cost performance on Bullet Safety Gym (LfN).** We report the mean and standard deviation across five random seeds. Higher reward and lower cost are preferred. These raw reward and cost results complement the Fβ results reported in the main paper, which directly evaluate the reward–safety trade-off.

Environment	Metric	SafeDICE	UNIQ	OSIL (LfN)	**PSIL (LfN)**
AntRun	Reward ↑	732.75 ± 10.44	269.83 ± 20.66	107.37 ± 19.46	708.06 ± 18.15
	Cost ↓	75.70 ± 6.50	7.03 ± 1.45	0.00 ± 0.00	68.08 ± 6.48
DroneRun	Reward ↑	395.60 ± 6.72	383.52 ± 59.86	307.50 ± 50.39	411.16 ± 16.72
	Cost ↓	9.39 ± 8.97	108.89 ± 9.89	38.24 ± 34.18	48.34 ± 13.84
AntCircle	Reward ↑	238.42 ± 23.04	82.22 ± 9.69	0.94 ± 0.87	239.88 ± 32.75
	Cost ↓	58.61 ± 11.31	16.51 ± 5.93	0.00 ± 0.00	55.30 ± 12.08
DroneCircle	Reward ↑	719.64 ± 11.92	77.10 ± 31.75	747.42 ± 16.49	794.81 ± 9.22
	Cost ↓	37.88 ± 2.32	12.33 ± 8.39	42.44 ± 4.55	56.22 ± 3.64

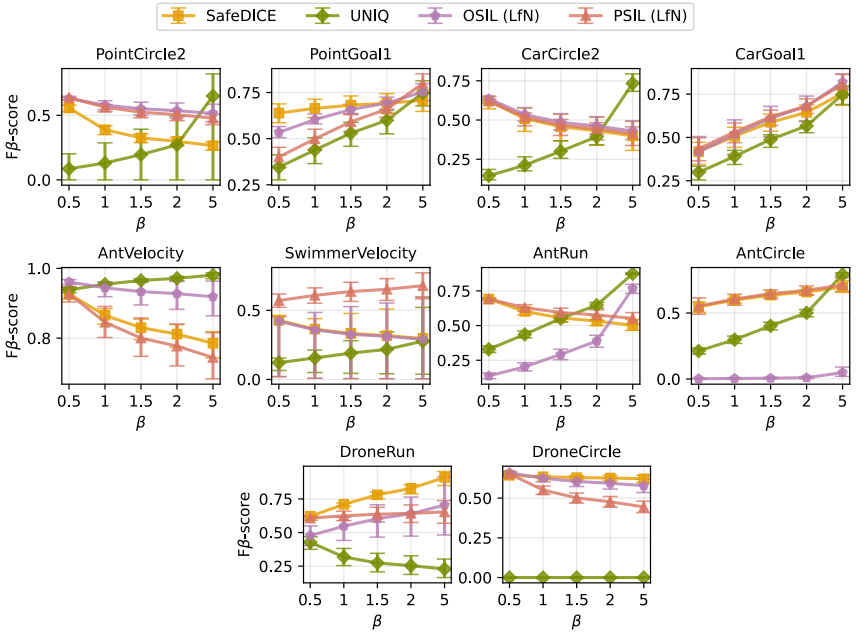

Figure 21: Comparison of the reward-safety trade-off across ten DSRL benchmark tasks in LfN setting. Each policy is trained for 1M gradient update steps and then evaluated using Fβ-score with $\beta \in \{0.5, 1, 1.5, 2, 5\}$. Smaller values of β place greater emphasis on reward, whereas larger values emphasize safety. Methods that maintain high Fβ-scores across different β values indicate a better reward-safety trade-off. Error bars denote 95% bootstrap confidence intervals.

L Impact of the Number of Non-Preferred Trajectories

We study the impact of number of non-preferred trajectories on our algorithm by varying $|\mathcal{D}_N| \in \{1, 3, 5, 10\}$ in the Point Circle2 and Drone Run environment. As shown in Figure 22, the performance gradually decreases with smaller $|\mathcal{D}_N|$, indicating the importance of non-preferred examples. We observe that when $|\mathcal{D}_N|$ is large, the non-preferred behaviors are well represented, we are able to recover low-cost behavior. However, as $|\mathcal{D}_N|$ decreases, the coverage of these non-preferred behaviors also decreases, making it harder to learn safe behavior.

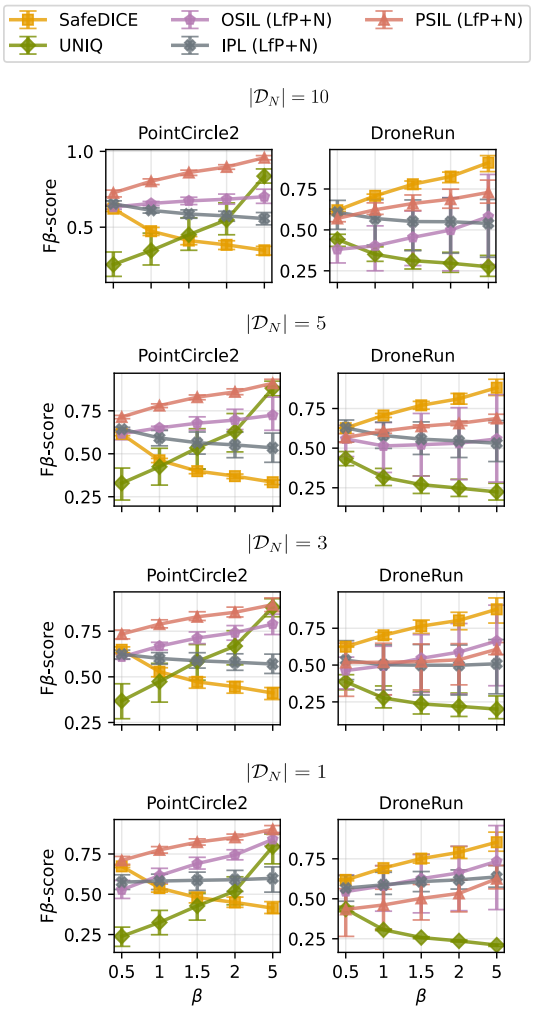

Figure 22: *Impact of Number of Non-Preferred Trajectories.*