Reinforcement Learning from Imperfect Demonstrations under Soft Expert Guidance

Xiaojian Ma^{*12}**, Mingxuan Jing**^{*1}**, Wenbing Huang**^{*1}**, Fuchun Sun**^{1}**, Chao Yang**^{1}**, Bin Fang**^{1}**, Huaping Liu**^{1}

[1]Beijing National Research Center for Information Science and Technology (BNRist),
State Key Lab on Intelligent Technology and Systems,
Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[2]Center for Vision, Cognition, Learning and Autonomy, Department of Computer Science, UCLA, CA 90095, USA
maxiaojian@ucla.edu, jmx16@mails.tsinghua.edu.cn
fcsun@tsinghua.edu.cn, hwenbing@126.com

Abstract

In this paper, we study *Reinforcement Learning from Demonstrations (RLfD)* that improves the exploration efficiency of Reinforcement Learning (RL) by providing expert demonstrations. Most of existing RLfD methods require demonstrations to be perfect and sufficient, which yet is unrealistic to meet in practice. To work on imperfect demonstrations, we first define an imperfect expert setting for RLfD in a formal way, and then point out that previous methods suffer from two issues in terms of optimality and convergence, respectively. Upon the theoretical findings we have derived, we tackle these two issues by regarding the expert guidance as a soft constraint on regulating the policy exploration of the agent, which eventually leads to a constrained optimization problem. We further demonstrate that such problem is able to be addressed efficiently by performing a local linear search on its dual form. Considerable empirical evaluations on a comprehensive collection of benchmarks indicate our method attains consistent improvement over other RLfD counterparts.

1 Introduction

Reinforcement Learning (RL) (Sutton and Barto 1998) enables robots to acquire skills by interacting with the environment. Despite the conspicuous advancements they have attained, typical RL methods suffer from the exploration issue that performing exploration over novel action-state trajectories is inefficient, and is not spontaneously guaranteed when the reward signals are sparse or incomplete. Thus, a fairly of approaches (Brys et al. 2015; Chemali and Lazaric 2015; Cederborg et al. 2015; Kang, Jie, and Feng 2018; Sun, Bagnell, and Boots 2018) have resorted to the combination of RL with expert demonstrations (containing action-state trajectories), giving rise to a new research vein that exploits expert demonstrations to help policy exploration of the agent. We refer this vein as Reinforcement Learning from Demonstrations (RLfD) in this paper.

Early RLfD methods enhance RL by either putting expert demonstrations into a replay buffer for value estimation (Hester et al. 2018; Večerík et al. 2017) or applying them to pre-train the policy in a supervised manner (Silver et al. 2016; Cruz Jr, Du, and Taylor 2017), both of which, however, simply regard demonstrations as data-augmentations without making full use of them during the policy optimization procedure. To address this weakness, modern RLfD approaches (Sun, Bagnell, and Boots 2018; Kang, Jie, and Feng 2018) absorb ingredients from Imitation Learning (Schaal 1997; Abbeel and Ng 2004; Ziebart et al. 2008; Ho and Ermon 2016; Jing et al. 2019; Yang et al. 2019), and encourage the agent to mimic the demonstrated behaviors when the environmental feedback is rare or even unavailable. Specifically, they reshape the native reward in RL by adding a demonstration-guided term to force expert-alike exploration.

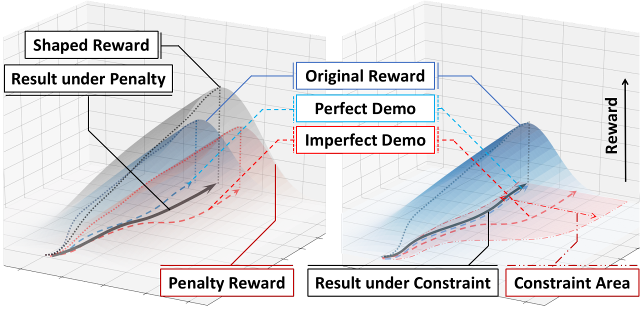

Figure 1: An overview of our RLfD method using soft constraint versus existing approaches using penalty departures. **Left**: In penalty method, agent seeks to maximize the shaped reward which may induce non-optimal solution. **Right**: Proposed soft constraint will guide the agent to explore towards areas with high reward without altering the optimality.

Whereas encouraging expert-alike actions does help in avoiding futile exploration, continuously enforcing such type of rewards during the whole learning phase is problematic if the provided demonstrations are imperfect. Here, the notion of *imperfect demonstrations* implies two senses: I. The quality of demonstrations is imperfect, which could be caused by data collection noise or intrinsically produced by the immaturity of the expert. II. The number of demonstrations is insufficient, which is due to the consuming resource and effort in collection. The imperfectness of demonstrations will potentially, if not always, make the convergence of the agent policy to be sub-optimal. As illustrated in Figure 1

[*]These two authors contributed equally. Correspondence to Fuchun Sun.

and non-strictly speaking, the learned agent policy by existing RLfD methods will converge to a point nearby the underlying expert policy. If the demonstrations/expert policies are imperfect, we have no guarantee to obtain better agent policy (or even have a potential detriment to the policy searching) by always minimizing its divergence to the expert behavior.

In this paper, we propose to conduct RL from imperfect demonstrations by applying expert guidance in a *soft* way. To illustrate our idea, let us revisit the example in Figure 1. We assume that the optimal agent policy still locates within a certain region around the imperfect expert policy (denoted by the red area in the **Right** of Figure 1), and once the agent policy is within this region, its optimization is only affected by the interaction with the environment and is no longer influenced by demonstrations. The intuition behind is that the expert demonstrations—even when they are imperfect—are able to characterize what actions are good in general but not precisely. Conventional RLfD methods fix the demonstration reward during the whole learning procedure and are not flexible enough to meet our requirement.

Towards our purpose, we reformulate the RLfD task as a constrained policy optimization problem (Altman 1999; Achiam et al. 2017; Tessler, Mankowitz, and Mannor 2019), where the goal is formulated by the native RL objective and the constraint is to bound the exploration region around the demonstrations under a certain threshold. By this formulation, the expert demonstrations regulate the agent policy updating only when the policy is outside the constraint region, which is consistent with our assumption mentioned above. Nevertheless, solving the constrained optimization problem is non-trivial. To tackle it effectively, we propose to search the optimal policy update for each step with a linearized subobjective. Through leveraging its dual form, we can significantly reduce the problem size and empowers the scalability to policy models with high-dimensional parameter space like neural networks. We provide more details in Sec. 3.

We summarize our contributions as follows.

- To the best of our knowledge, we are the first to formulate RLfD as a constrained optimization problem, by which we are able to make full use of imperfect demonstrations in a soft and also more effective manner.
- We develop an efficient method to solve the proposed constrained optimization problem with scalable policy model like deep neural networks.
- With imperfect demonstrations, our method achieves consistent improvement over other RLfD counterparts on several challenging physical control benchmarks.

The rest of paper is organized as follows: In Sec. 2, we first provide necessary notations and preliminaries about the subject of RLfD. Then our proposed method will be detailed in Sec. 3 with analysis and efficient implementation. The discussion on some related research will be included in Sec. 4. Finally, experimental evaluations will be demonstrated in Sec. 5.

2 Preliminaries

Notations. For modeling the action decision process in our context, a standard Markov decision process (MDP) (Sutton and Barto 1998) $(\mathcal{S}, \mathcal{A}, r, \mathcal{T}, \mu, \gamma)$ is considered, where $\mathcal{S}$ and $\mathcal{A}$ denotes the space of feasible states and actions respectively, $r(s, a) \rightarrow \mathbb{R}$ is the reward function, $\mathcal{T}(s'|s, a)$ and $\mu(s)$ represent the transition probability and initial state distribution and $\gamma \in (0, 1)$ is the discount factor. A stochastic policy $\pi(a|s) : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ maps state into action distribution. A trajectory ζ is given by the sequence of state-action pairs $\{(s_0, a_0), (s_1, a_1), ...\}$.

Occupancy measure. The concept of occupancy measure (Puterman 1994; Syed, Bowling, and Schapire 2008) defined below characterizes the distribution of the state-action pairs within the exploration trajectories when policy π is executed, which will be useful in the following analysis.

Definition 1 (Occupancy Measure)**.** *Given a stationary policy π, let $\rho_\pi(s) : \mathcal{S} \rightarrow \mathbb{R}$ and $\rho_\pi(s, a) : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ denote the density of the state distribution and the joint distribution for state and action under the policy π,*

$$\begin{aligned} \rho_\pi(s) &\triangleq \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi) \\ \rho_\pi(s, a) &\triangleq \rho_\pi(s)\pi(a|s). \end{aligned} \tag{1}$$

We name $\rho_\pi(s, a)$ as occupancy measure of policy π.

Formulation of RLfD. The objective of RL is to maximize the cumulative expected (discounted) return along the whole decision procedure $\eta(\pi) = \mathbb{E}_\pi[\sum_t^\infty \gamma^t r(s_t, a_t)]$, given current action policy π (Sutton and Barto 1998). While RLfD enhances RL with providing a set of demonstrated trajectories $\mathcal{D} = \{\zeta_0, \cdots\}$ draw from a referred expert with policy π_E as an extra guidance other than reward. Such expert data can be useful notably when the environmental feedback is sparse or delayed (Pathak et al. 2017), in which the agent may suffer from ineffective explorations since positive feedback could rarely occur.

RLfD with penalty departures. Some previous research(Brys et al. 2015; Kang, Jie, and Feng 2018) suggest exploring towards the area that frequently visited by expert policy π_E, because it may provide a higher and denser return that agent can benefit from. As it mentioned above, such visiting frequency is essentially characterized by expert's occupancy measure. Intuitively, we can leverage the distribution discrepancy between the occupancy measure of expert and agent as an extra feedback signal to encourage this exploration behavior, which gives us the following objective

$$\min_\pi \mathcal{L}_\pi = -\eta(\pi) + \lambda \cdot \mathbb{D}\left[\rho_\pi(s, a) \| \rho_E(s, a)\right], \tag{2}$$

where $\mathbb{D}(\cdot\|\cdot)$ and $\rho_E(s, a)$ depict any discrepancy measure and expert's occupancy measure respectively, λ is an adjustable weight. We refer (2) as *RLfD with penalty departures* in the following context since the discrepancy is introduced as a penalty function upon the original objective of RL and can be approximated through expert demonstrations.

3 Methodology

In this section, we will first introduce the new setting of *imperfect* expert for RLfD and emphasize the optimality and convergence issues in the penalty method, which essentially motivates our approach to employ expert guidance as a soft constraint instead. We also demonstrate that such constrained optimization problem can be solved efficiently by performing a local linear search on its dual form, maintaining its scalability to complex policy model like deep neural networks. Finally, we provide a practical implementation of our method.

3.1 Expert Guidance as a Soft Constraint: Towards RLfD with an Imperfect Expert

We now illustrate the imperfect setting for RLfD. As it mentioned in Sec. 1, the imperfectness here is raised from two facets: **quality** and **amount** of available demonstrations. Here we first focus on the quality, and the issue on the amount of demonstrations will be discussed later. Compared to the perfect expert setting that assumed the expert policy has already maximized the expected return (Brys et al. 2015; Kang, Jie, and Feng 2018), an imperfect expert employs a policy that still not converge to an expected local optimum *w.r.t.* the considered RL objective. Without loss of generality, an imperfect expert, can be defined as follows.

Definition 2 (Imperfect Expert Policy)**.** *Denoting π_{θ^+} and π_{θ^-} as perfect and imperfect expert policies respectively. π_{θ^-} either attains local optimum with a lower return than π_{θ^+} or does not belong to any local optima.*

$$\pi_{\theta^+} \in \left\{\pi : \arg\max_{\pi} \eta(\pi) \;\; \mathit{AND} \;\; \frac{\partial \eta(\pi_{\theta^+})}{\partial \theta^+} = 0\right\}$$

$$\pi_{\theta^-} \in \left\{\pi : \left\{ \tfrac{\partial \eta(\pi_{\theta^-})}{\partial \theta^-} = 0 \;\; \mathit{AND} \;\; \eta(\pi_{\theta^-}) < \eta(\pi_{\theta^+}) \right\} \right.$$
$$\left. \mathit{OR} \;\; \left\{ \tfrac{\partial \eta(\pi_{\theta^-})}{\partial \theta^-} \neq 0 \right\} \right\},$$

where $\eta(\pi)$ is the objective of currently considered RL task.

The penalty method presented in Sec. 2 works comparably well when expert is optimal (Brys et al. 2015; Kang, Jie, and Feng 2018). However, optimizing the composite sum of two parts in (2) under imperfect expert setting is problematic, as it may alter the optimality and induces no convergence guarantee for the original RL objective. The following propositions illustrate this issue formally.

Proposition 1. *Denoting $\pi_{\theta^\star} = \arg\max_\pi \eta(\pi)$ as the optimal policy under the given RL objective $\eta(\pi)$. Then for the additional distribution discrepancy term $\mathbb{D}\left[\rho_\pi \| \rho_E\right]$ in* (2), $\frac{\partial \mathbb{D}[\rho_{\pi_\theta} \| \rho_{\pi_{\theta^+}}]}{\partial \theta}\Big|_{\theta=\theta^\star} = 0$. *But when an imperfect expert π_{θ^-} is adopted, this result does not hold under certain conditions.*

This proposition presents an intuitive result that the optimal policy for a given RL task can't always be an optimum of the additional discrepancy term in (2) under imperfect demonstrations. We will further show that this may make (2) converge to a solution that is non-optimal for the original RL problem.

Proposition 2. *When the penalty method* (2) *under imperfect demonstrations converges to a local optimum π_θ, it can't always be the optimal solution for the original RL objective.*

$$\frac{\partial \mathcal{L}_{\pi_\theta}}{\partial \theta} = 0 \not\Rightarrow \pi_\theta = \arg\max_{\pi} \eta(\pi).$$

While under the same certain condition as Proposition 1, we can obtain an even stronger conclusion

$$\frac{\partial \mathcal{L}_{\pi_\theta}}{\partial \theta} = 0 \Rightarrow \eta(\pi_\theta) < \max_{\pi} \eta(\pi).$$

The two propositions above imply that, under the imperfect setting, the additional penalty term will substantially change the optimization landscape of original RL problem and may induce convergence to a non-optimal solution. Although it can offer positive guidance in the early training phase, it will soon become misleading and prevent the policy from attaining higher return. To tackle this issue, we propose to transform the distribution discrepancy penalty term into a constraint instead. This intuition is actually based on the following observation:

Proposition 3. *There exists a bounded tolerance factor d such that the optimal policy $\pi_{\theta^\star}$ always stay within an area closer to the demonstrations specified by d, even when the demonstrations are drawn from an imperfect expert.*

$$\exists d \in [0, \infty), \mathbb{D}\left[\rho_{\pi_{\theta^\star}} \| \rho_{\pi_{\theta^-}}\right] \leqslant d, \pi_{\theta^\star} = \arg\max_{\pi} \eta(\pi).$$

From the perspective of optimization, it suggests that using constraint could better fit the imperfect expert setting by two reasons. **1. Optimality.** Refer to Proposition 3, given a proper tolerance d, once the optimal policy satisfies the constraint, the new constrained optimization problem will share the same optimal solution with the original RL problem. **2. Convergence.** The constraint only affects policy update when it is not satisfied. Therefore, when the policy improves to a certain extent, *i.e.* stays within the constraint, it will only learn from the original reward feedback and finally converge to the optimality of the original RL problem. As a conclusion, compared to the penalty method, the constraint method can leverage the imperfect demonstrations for guiding the policy while eliminating their side effects in optimization, thus can work better with imperfect experts.

For another facet of imperfectness, *i.e.* amount, as the expert data is mainly introduced for computing the distribution discrepancy in our context, the issue of insufficient amount of demonstrations will essentially rely on the estimation error to the discrepancy, which may induce bias to policy update especially when the gradient step is relatively large. We refer to the idea of local policy search (Kakade 2002; Kakade and Langford 2002) to alleviate this issue by making conservative gradient step instead with an auxiliary constraint on the change of Kullback-Leibler (KL) divergence of the updated policy.

Overall optimization problem. By combining the discrepancy constraint and local policy search, the eventual optimization problem (k-th step) with imperfect expert π_{θ^-} is

$$\begin{aligned} \theta_{k+1} = \arg\max_{\theta} \;\; & \eta(\pi_{\theta_k}) \\ \text{s.t.} \;\; & \mathbb{D}\left[\rho_{\pi_{\theta_k}}(s,a) \| \rho_{\pi_{\theta^-}}(s,a)\right] \leqslant d_k \\ & \mathbb{D}_{\mathrm{KL}}\left[\pi_{\theta_k}(a|s) \| \pi_{\theta_{k+1}}(a|s)\right] \leqslant \delta, \end{aligned} \tag{3}$$

where δ is the tolerance of the KL constraint. The remaining issue now is how to determine the tolerance factor d_k for the discrepancy constraint in each step. To avoid hand-crafting this parameter on different tasks and demonstrations, we apply a simple annealing strategy on d_k to realize a **soft** constraint as it can adapt along with the improvement of policy, comparing to a fixed tolerance. Specifically, we adopt the following update rule for d_k

$$d_{k+1} \leftarrow d_k + d_k \cdot \epsilon, \tag{4}$$

where ϵ is the annealing factor. We will further demonstrate the advantage on adopting a soft constraint and the strategy on hyper-parameter choosing in our empirical evaluations in Sec. 5.4.

3.2 Solving with Scalable Policy Models

We've shown the issues of the penalty method for RLfD when the expert data is imperfect, and therefore motivated our new approach that reformulates it as a constrained policy optimization problem (3). Nevertheless, solving it accurately can be rather challenging due to: **1. Feasibility**, it may be difficult to find a feasible solution with the two constraints. **2. Scalability**, for policies that are characterized by a model with high-dimensional parameter space, *i.e.* neural networks, the computation cost of the new constraint will become unaffordable. To this end, we propose to approximately solve it by linearizing around π_{θ_k} at each optimization step. Denoting the gradient of the objective as g, the current discrepancy at θ_k as d_{θ_k} and its gradient as b, the Hessian matrix of the KL-divergence as H[1], the linear approximation to (3) is

$$\begin{aligned} \theta_{k+1} = \arg\max_{\theta} \quad & g^T(\theta - \theta_k) \\ \text{s.t.} \quad & b^T(\theta - \theta_k) + d_{\theta_k} \leqslant d_k \\ & \frac{1}{2}(\theta - \theta_k)^T H (\theta - \theta_k) \leqslant \delta. \end{aligned} \tag{5}$$

The approximated optimization problem above is convex as H is always positive semi-definite (Schulman et al. 2015). Therefore, compared to its original form (3), a feasible solution can be found more easily using duality. In particular, given λ and ν as the Lagrange multipliers for KL-divergence and discrepancy constraints, a corresponding dual to (5) can be written as

$$\max_{\substack{\lambda \geq 0 \\ \nu \geq 0}} -\frac{1}{2\lambda}(g^T u + 2\nu b^T u + \nu^2 b^T r) - \nu c - \lambda\delta, \tag{6}$$

where $u = H^{-1}g$, $r = H^{-1}b$, $c = d_k - d_{\theta_k}$. Since the number of variables in this dual problem is much less than the dimension of θ, the computation cost will also be much less than solving (3). A closed-form solution of optimal solution $\lambda^\star$, $\nu^\star$ can be derived by firstly obtaining and substituting $\nu^\star$, then discussing the sub-case and finally gets $\lambda^\star$. Suppose we have the optimal solution $\lambda^\star$, $\nu^\star$ of this dual problem, the solution to the primal one will be

$$\theta^\star_{k+1} = \theta_k - \frac{1}{\lambda^\star}(u + r\nu^\star). \tag{7}$$

Algorithm 1 RLfD with a Soft Constraint

```
Input: Imperfect expert demonstrations $\mathcal{D}_E = \{\zeta_i^E\}$, initial policy $\pi_{\theta_0}$, initial constraints tolerance $d_0$, $\delta$, annealing factor $\epsilon$, maximal iterations $N$.
for k = 0 to $N$ do
    Sample roll-out $\mathcal{D}_\pi$ with $\pi_{\theta_k}$.
    Estimate $\hat{g}$, $\hat{b}$, $\hat{H}$ with samples from $\mathcal{D}_E$ and $\mathcal{D}_\pi$.
    if the optimization problem (5) is feasible then
        Solve the dual problem (6) to obtain $\lambda^*$, $\nu^*$.
        Compute update step proposal $\Delta\theta$ as (7).
        Update the policy by backtracking line-search along $\Delta\theta$ to ensure the satisfaction of constraints.
    else
        Update the policy via the recovery objective (9).
    end if
    Annealing the tolerance $d_k$: $d_{k+1} \leftarrow d_k + d_k \cdot \epsilon$.
end for
```

When there is at least one feasible point within the KL constraint (the trust region), we can update the policy parameter θ by solving the dual for $\lambda^\star$ and $\nu^\star$ (7). However, due to the initialization and approximation error, the proposed update rule may sometimes not satisfy the constraints in (3), especially at the beginning of optimization. In the next section, we will provide more details on ensuring the feasibility.

3.3 Implementation Details

The choice of discrepancy measure. In RLfD, as we can only access the samples (demonstrations) from the expert policy and its occupancy measure, we adopt the non-parametric distance metric MMD (Gretton et al. 2007; Sriperumbudur et al. 2008; Gretton et al. 2012) as the discrepancy measure. The value and gradient *w.r.t.* policy parameters of MMD can be easily computed with demonstrations and agent roll-outs. Moreover, we use the characteristic Gaussian kernel to ensure the following property

$$\mathrm{MMD}[p, q] = 0 \Leftrightarrow p = q, \tag{8}$$

where p, q denote two distributions. This property can help alleviate the inconsistency between minimizing discrepancy and morphing distributions within the discrepancy constraint and improve the optimization (Smola et al. 2007).

Feasibility issue. The major crux that accounts for the feasibility issue when solving (3) can be twofold. One lies in the beginning phase. As the parameter θ is usually randomly initialized, it may induce infeasibility when the optimization just starts. We propose a recovery strategy that transforms the constraint into an objective to eliminate this issue.

$$\theta^\star = \arg\min_{\theta} \mathbb{D}\left[\rho_{\pi_\theta}(s, a) \| \rho_{\pi_{\theta^-}}(s, a)\right]. \tag{9}$$

Another source of infeasibility comes from (7). The update rule may not satisfy the constraints due to the approximation error. To this end, we apply a backtracking line-search along $\Delta\theta = -{\lambda^\star}^{-1}(u + r\nu^\star)$ to ensure the constraint satisfaction. To further reduce the computation cost, we also adopt the conjugate gradient method like (Schulman et al. 2015) to approximately compute the inverse of H and its products.

The algorithm detail is summarized in Algorithm 1.

[1] The KL constraint should be approximated via second-order expansion since its first order gradient is zero at $\pi_\theta = \pi_{\theta_k}$.

4 Discussion

In this section, we will discuss some relevant research on RLfD, and demonstrate how they connect to our method.

Pre-train with demonstrations. A straight-forward solution for combining demonstrations in RL will be pre-training agent policy with expert data via imitation learning, *e.g.* behaviour cloning (Schaal 1997; Atkeson and Schaal 1997), then proceeding with normal RL (Silver et al. 2016; Cruz Jr, Du, and Taylor 2017). The first step is similar to our constrained optimization approach under unsatisfied constraints when the optimization starts, sometimes even have better performance at the beginning. However, this method cannot guarantee the exploration quality in the later RL step; thus the subsequent training can still suffer from poor sample efficiency in the case with large exploration space and sparse feedback.

Penalty with other discrepancy measures. There is also some research on investigating different discrepancy measures for RLfD with penalty departures (Brys et al. 2015; Kang, Jie, and Feng 2018). Notable recent research is **POfD** (Kang, Jie, and Feng 2018), which proposed to leverage Generative Adversarial Networks (GANs) (Goodfellow et al. 2014) to evaluate the discrepancy between the occupancy measure of expert and agent. In our comparative evaluations, it demonstrates comparative performances than baseline that employs MMD as penalty departures. However, this method requires an extra parameterized model (discriminator) and training procedure (adversarial training), which substantially increase the difficulty of convergence.

Penalty with annealing. In Sec. 3.1, we've mentioned that our constraint method adopts an annealing strategy to select the constraint factor adaptively. Since our method would expect the optimal policy to stay within the constraint, annealing is more practical than manually specifying a fixed factor for different task and demonstrations. Similarly, this strategy is also applicable to the factor λ in penalty method (2) for suppressing the side effect of imperfect demonstrations. However, we should notice that annealing can only partly alleviate this impact before λ becomes zero. While in our approach, only the original RL objective is being optimized once the constraint with imperfect expert data is satisfied. In fact, our empirical results in Sec. 5.2 indicate penalty with annealing does perform advantageously than pure penalty method in some evaluated tasks, but there is still a significant gap to our approach using soft constraint.

5 Experiments

For the experiments below, we aim at investigating the following questions:

1. Under the same imperfect expert settings, can our method attains better performative results versus the counterparts that do not employ demonstrations as a soft constraint?
2. How can the different settings of imperfect expert data, *i.e.* quality and amount, affect the performances of our method and baselines?
3. What is the key ingredient in our method that introduces better empirical results?

To answer the first question, we evaluate our method against several baselines on six physical control benchmarks (Duan et al. 2016; Brockman et al. 2016), ranging from low-dimensional classical control to challenging high-dimensional continuous robotic control tasks. Regarding the second question, we conduct ablation analysis on the quality and amount of demonstrations, respectively. We test and contrast the performances of our method and two representative baselines (Pre-train (Silver et al. 2016) and POfD (Kang, Jie, and Feng 2018)) on these different imperfect expert setting. Finally, we explore another ablation analysis on the core component in our method, *i.e.* soft constraint to address the last question.

5.1 Settings

To simulate the sparse reward conditions using existing control tasks in Gym, we first propose several reward sparsification methods with details as follows[2]:

- **S1**: Agent receives reward $+1$ when it reaches a specific terminal state; otherwise, no reward will be provided.
- **S2**: Agent receives reward $+1$ when has already moved towards a certain direction for some distance.
- **S3**: Agent receives reward $+1$ when its last pole is higher than a given height. Only applied to *DoublePendulum* task.

We train expert policies (namely *perfect* experts, shown as **Expert**) for each tested tasks with PPO (Schulman et al. 2017) based on the exact reward, and select policies learned meanwhile (namely *imperfect* experts, shown as **Demo**), record only **one** trajectory as the imperfect demonstrations. To make the comparisons fairer, the policies of all the methods and tasks are parameterized by the same neural network architecture with two hidden layers (300 and 400 units each) and tanh activation functions. All the algorithms are evaluated within the fixed amount of environment steps. And for every single task, we run each algorithm over five times with different random seeds.

5.2 Comparative Evaluations

In comparative evaluations, we carry out several RLfD baselines, including Pre-train (Silver et al. 2016) and POfD (Kang, Jie, and Feng 2018). In particular, we introduce another two baselines of penalty method[3] with MMD as discrepancy measure, denoted by Penalty and Penalty + Ann., and the later one also employ an annealing strategy described in Sec. 4. We also run two non-RLfD baselines PPO and MMD-Imitation (denoted as MMD-IL) to verify the reward sparsification and the imperfect expert setting respectively. PPO will run with the sparse reward while MMD-IL will directly optimize the objective defined in (9) with provided imperfect demonstrations. In Figure 2, the solid curves correspond to the mean reward, and the shaded region represents the variance over five times. The results of our compar-

[2] The presented results are still evaluated in the original exact reward defined in (Brockman et al. 2016).

[3] POfD also belongs to penalty method.

Table 1: Comparative results (with only 1 imperfect demonstration). All results are measured in the original exact reward.

	MountainCar	DoublePendulum	Hopper	Walker2d	HalfCheetah	Ant
$\mathcal{S}$ / $\mathcal{A}$	$\mathbb{R}^4$ / $\{0,1\}$	$\mathbb{R}^{11}$ / $\mathbb{R}^1$	$\mathbb{R}^{11}$ / $\mathbb{R}^3$	$\mathbb{R}^{17}$ / $\mathbb{R}^6$	$\mathbb{R}^{17}$ / $\mathbb{R}^6$	$\mathbb{R}^{111}$ / $\mathbb{R}^8$
Setting / Demo	**S1** / 81.25	**S3** / 1488.28	**S2** / 969.71	**S2** / 1843.75	**S2** / 2109.80	**S2** / 1942.05
PPO	-0.74±9.61	302.77±37.09	17.09±13.54	1.54±5.75	978.84±665.61	-2332.95±2193.85
MMD-IL	82.99±4.57	218.43±13.72	118.66±0.38	8.88±6.07	161.74±219.85	967.83±0.87
Pre-train	83.35±6.32	8928.79±388.61	1356.47±470.43	2607.38±301.94	3831.96±150.30	-5377.25±1682.56
POfD	45.01±28.16	628.47±69.36	32.13±24.23	-1.48±0.03	2801.59±66.03	-68.59±19.17
Penalty	-120.29±48.30	1902.95±210.41	1225.03±296.52	286.23±12.46	1517.68±35.85	-3711.12±794.97
Penalty + Ann.	79.00±1.04	1671.78±108.80	1220.10±112.74	282.00±6.70	2592.94±870.04	-116.89±88.01
Ours	**83.46±1.42**	**9331.40±5.95**	**2329.89±125.85**	**3483.78±269.59**	**4106.69±95.47**	**2645.58±118.55**

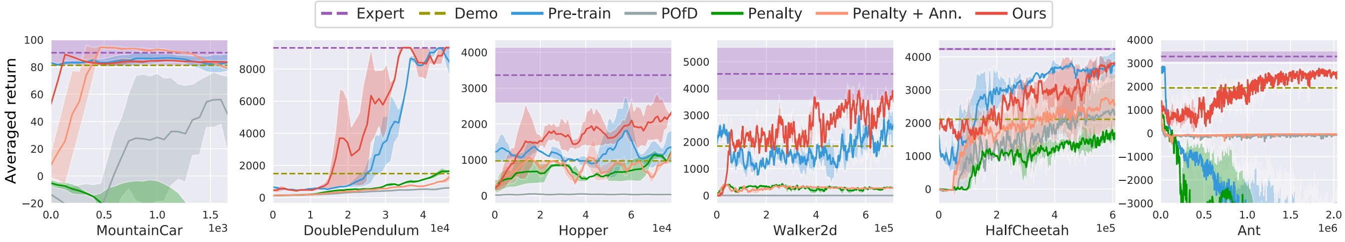

Figure 2: Learning curves of our method versus baselines under challenging robotic control benchmark. For each experiment, **a step represents one interaction with the environment**. The number of steps could be variant in different figures.

ative evaluations are summarized in Table 1, which averaged 50 trials under the learned policies.

The results overall read that our method achieves comparable performances with the baselines on relatively simple tasks (such as *MountainCar*) and outperforms them with a large margin on difficult tasks (such as *Hopper*, *Walker2d* and *Ant*). During policy optimization, our method can converge faster than other RLfD counterparts as well as obtains better final results. Comparing with the strong baseline of Pre-train, we can see that although convergence efficiency of proposed method during the early phase of training may not have significant advantages, but as it continues, the performance of our method can be improved persistently like *Hopper*(+973.42) and *Walker2d*(+876.40), while Pre-train struggles on achieving higher return, which demonstrates that our method could benefit more from the exploration guidance offered by the soft constraint during the whole policy optimization procedure than by only imitating at the beginning.

On the other hand, we also find that our algorithm exhibits a more stable and efficient behavior over all the baselines using the penalty method. From the learning curve and numerical results, it can be seen that adopting penalty with imperfect demonstrations will induce a noisy and misleading gradient update, which will prevent the performances from improving further while our method with a soft constraint will not suffer from this. This essentially accounts for the performance gap between our method and all baselines with penalty departures. Moreover, the complex training strategies and auxiliary model in POfD also leads to unstable and inefficient training across different tasks and environment specifications.

From the results of PPO and MMD-Imitation, the experiment settings of reward sparsification and imperfect demonstrations can be verified. As it illustrates, under sparse environmental feedback, pure PPO fails to find an optimal policy on most of the tested tasks, which indicates the impact of ineffective exploration. While with few imperfect demonstrations, MMD-Imitation also cannot learn promising policies. It suggests that combining the demonstrations and environmental feedback would be essential for the designated tasks. Furthermore, as similar MMD-Imitation update may happen in our method when the optimization just starts (mentioned in Sec. 3.3), these results also show how can our method benefit from the follow-up solving of the constrained optimization problem.

5.3 Ablation Analysis I: Sensitivity to Demonstrations

The results presented in the previous section suggest that our proposed method outperforms other RLfD approaches on several challenging tasks. We're now interested in whether these advantages still hold when the demonstration setting changes. We will compare our method and baselines on demonstrations with different amounts and quality respectively to show how can they affect the performative results.

Demonstrations with different amounts. We select six groups of demonstrations with different amounts from 50 to 5000 for comparison on the *HalfCheetah* task. Notice the

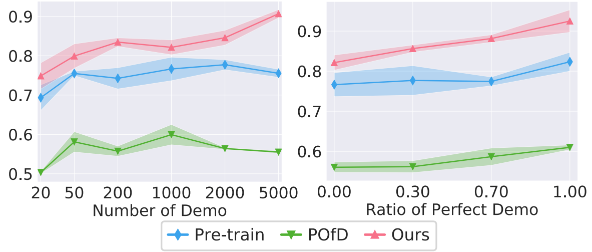

Figure 3: Results on *HalfCheetah* task with different imperfect expert setting. **Left**: Different number of state-action pairs; **Right**: Different level of imperfectness.

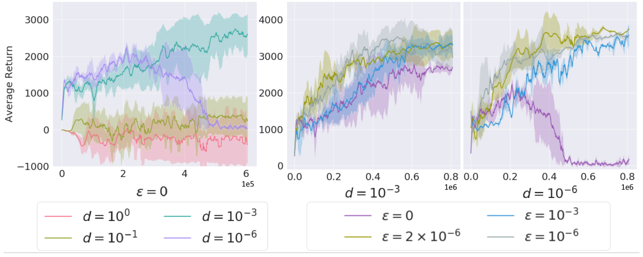

Figure 4: Learning curves over on *HalfCheetah* task. **Left**: ablation study about different tolerance factor d; **Right**: sensitivity of choosing fixed or annealing strategy of tolerance.

comparative experiments in Sec.5.2 are conducted with one trajectory with 1000 state-action pairs as demonstrations. The corresponding results are plotted in the **Left** of Figure 3. The results read that our method performs advantageously than the baselines on these demonstration settings, and the performance gap is also getting larger as the number of demonstrations increases. On the other hand, the results could benefit from more demonstrations in a certain range for all the methods, while our method can be more robust when the demonstrations become fewer.

Demonstrations with different qualities. We emulate the demonstrations of different qualities by mixing the demonstrated data from perfect (**Expert**) and imperfect (**Demo**) policies with different ratios. The **Right** of Figure 3 presents the results of our method and baselines with these demonstrations. It implies that the quality of demonstrations will significantly affect the performances of all the evaluated methods, and expert data with high quality can facilitate policy optimization to a certain extent. We can also see that our method overall outperforms the two counterparts even though the expert data becomes perfect (by setting the ratio to 1.00), indicating that our constraint-based method can exploit the expert data more efficiently than other methods based on penalty departures or pre-training.

5.4 Ablation Analysis II: Sensitivity to Constraint Tolerance

Now we will further investigate how can the design of the core soft constraint affect the performative results of our method. More specifically, we're interested in the tolerance factor d. By varying the initial value of d and annealing strategies (namely, different annealing factor ϵ), we will explore the sensitivity of our algorithm regarding them.

Different tolerance. We design four groups of parameters for the ablation experiments on the tolerance choosing in *HalfCheetah* task, where the annealing mechanism is disabled by setting ϵ fixed at zero, and choose initial tolerance d_0 from $\{10^0, 10^{-1}, 10^{-3}, 10^{-6}\}$. The learning curves are plotted in **Left** of Figure 4. As the results demonstrate, when given relatively large tolerance, the exploration reference from demonstrations will not work as the constraint almost does not affect policy optimization. In contrast, a too-small tolerance will hurt the final performance when the demonstrations are imperfect. Therefore, hand-crafting the tolerance for the constraint can be difficult, and an automatic adjustment with the annealing mechanism should be adopted.

Fixed vs. Annealing tolerance. In the previous experiment, we mention the importance of annealing of tolerance. Now we explore the advantages of annealing mechanism quantitatively in *HalfCheetah* task. Since our annealing is to enlarge the tolerance along training, we only choose two not-too-large initial tolerances d_0 from $\{10^{-3}, 10^{-6}\}$, and select the annealing factor ϵ from $\{0, 2 \times 10^{-3}, 10^{-3}, 10^{-6}\}$. Corresponding learning curves are shown in **Right** of Figure 4. We can see that the performances of our method with an annealing tolerance are overall better than with a fixed one (simply by setting ϵ as zero). Moreover, when the annealing factor ϵ is set properly, the performance of our method is not sensitive to the minor changes of ϵ as the results of different factors are almost at the same level. This further demonstrates the robustness of our proposed method.

6 Conclusion

In this paper, we investigate the problem of RLfD with imperfect expert data. Compared to existing RLfD problem setting, this new setting does not require the expert to be optimal, which can be more practical for real-world demonstrators like a human. We show that current penalty based RLfD methods will suffer from the issue of optimality and convergence when being applied to the setting of imperfect experts both theoretically and empirically. To this end, we propose to employs the expert data as a soft constraint and reformulate RLfD as a constrained policy optimization problem to narrow the negative impact of the imperfectness. We also provide an efficient learning algorithm for solving the challenging constrained optimization problem with scalable policy model like neural networks. Experiments on physical control benchmarks demonstrate the effectiveness of our proposed method over other RLfD counterparts. While we still assume the expert data to be collected from the same domain as the current conducted task, further exploration on combining our work with representation learning to enable learning with demonstrations across different domains could be a new direction of future work.

Acknowledgment

This research was funded by National Science Foundation of China (Grant No.91848206). It was also partially supported by the National Science Foundation of China (NSFC) and the German Research Foundation (DFG) in project Cross Modal Learning, NSFC 61621136008/DFG TRR-169. We would like to thank Dr. Boqing Gong and Dr. Tao Kong for their generous help and insightful advice.

References

Abbeel, P., and Ng, A. Y. 2004. Apprenticeship learning via inverse reinforcement learning. In *International conference on Machine learning (ICML)*.

Achiam, J.; Held, D.; Tamar, A.; and Abbeel, P. 2017. Constrained policy optimization. In *International Conference on Machine Learning (ICML)*.

Altman, E. 1999. *Constrained Markov decision processes*, volume 7. CRC Press.

Atkeson, C. G., and Schaal, S. 1997. Robot learning from demonstration. In *International Conference on Machine Learning (ICML)*.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. Openai gym.

Brys, T.; Harutyunyan, A.; Suay, H. B.; Chernova, S.; Taylor, M. E.; and Nowé, A. 2015. Reinforcement learning from demonstration through shaping. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Cederborg, T.; Grover, I.; Isbell, C. L.; and Thomaz, A. L. 2015. Policy shaping with human teachers. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Chemali, J., and Lazaric, A. 2015. Direct policy iteration with demonstrations. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Cruz Jr, G. V.; Du, Y.; and Taylor, M. E. 2017. Pre-training neural networks with human demonstrations for deep reinforcement learning. *arXiv preprint arXiv:1709.04083*.

Duan, Y.; Chen, X.; Houthooft, R.; Schulman, J.; and Abbeel, P. 2016. Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning (ICML)*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Gretton, A.; Borgwardt, K.; Rasch, M.; Schölkopf, B.; and Smola, A. J. 2007. A kernel method for the two-sample-problem. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Gretton, A.; Borgwardt, K. M.; Rasch, M. J.; Schölkopf, B.; and Smola, A. 2012. A kernel two-sample test. *Journal of Machine Learning Research (JMLR)*.

Hester, T.; Vecerik, M.; Pietquin, O.; Lanctot, M.; Schaul, T.; Piot, B.; Horgan, D.; Quan, J.; Sendonaris, A.; Osband, I.; et al. 2018. Deep q-learning from demonstrations. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Ho, J., and Ermon, S. 2016. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Jing, M.; Ma, X.; Huang, W.; Sun, F.; and Liu, H. 2019. Task transfer by preference-based cost learning. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Kakade, S., and Langford, J. 2002. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning (ICML)*.

Kakade, S. M. 2002. A natural policy gradient. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Kang, B.; Jie, Z.; and Feng, J. 2018. Policy optimization with demonstrations. In *International Conference on Machine Learning (ICML)*.

Pathak, D.; Agrawal, P.; Efros, A. A.; and Darrell, T. 2017. Curiosity-driven exploration by self-supervised prediction. In *International Conference on Machine Learning (ICML)*.

Puterman, M. L. 1994. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Schaal, S. 1997. Learning from demonstration. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015. Trust region policy optimization. In *International Conference on Machine Learning (ICML)*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of go with deep neural networks and tree search. *nature* 529(7587):484.

Smola, A.; Gretton, A.; Song, L.; and Schölkopf, B. 2007. A hilbert space embedding for distributions. In *International Conference on Algorithmic Learning Theory (ALT)*.

Sriperumbudur, B. K.; Gretton, A.; Fukumizu, K.; Lanckriet, G.; and Schölkopf, B. 2008. Injective hilbert space embeddings of probability measures. In *Annual Conference on Learning Theory (COLT)*.

Sun, W.; Bagnell, J. A.; and Boots, B. 2018. Truncated horizon policy search: Combining reinforcement learning and imitation learning. In *International Conference on Learning Representations (ICLR)*.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning: An introduction*. MIT Press.

Syed, U.; Bowling, M.; and Schapire, R. E. 2008. Apprenticeship learning using linear programming. In *International Conference on Machine Learning (ICML)*.

Tessler, C.; Mankowitz, D. J.; and Mannor, S. 2019. Reward constrained policy optimization. In *International Conference on Learning Representations (ICLR)*.

Večerík, M.; Hester, T.; Scholz, J.; Wang, F.; Pietquin, O.; Piot, B.; Heess, N.; Rothörl, T.; Lampe, T.; and Riedmiller, M. 2017. Leveraging demonstrations for deep reinforcement learning on robotics problems with sparse rewards. *arXiv preprint arXiv:1707.08817*.

Yang, C.; Ma, X.; Huang, W.; Sun, F.; Liu, H.; Huang, J.; and Gan, C. 2019. Imitation learning from observations by minimizing inverse dynamics disagreement. In *Advances in Neural Information Processing Systems 32*. 239–249.

Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; and Dey, A. K. 2008. Maximum entropy inverse reinforcement learning. In *AAAI Conference on Artificial Intelligence (AAAI)*.