LAGRANGIAN GENERATIVE ADVERSARIAL IMITATION LEARNING WITH SAFETY

Anonymous authors
Paper under double-blind review

ABSTRACT

Imitation Learning (IL) merely concentrates on reproducing expert behaviors and could take dangerous actions, which is unbearable in safety-critical scenarios. In this work, we first formalize a practical task of safe imitation learning (Safe IL), which has been long neglected. Taking safety into consideration, we augment Generative Adversarial Imitation Learning (GAIL) with safety constraints and then relax it as an unconstrained saddle point problem by utilizing a Lagrange multiplier, dubbed LGAIL. Then, we apply a two-stage optimization framework to solve LGAIL. Specifically, a discriminator is firstly optimized to measure the similarity between the agent-generated state-action pairs and the expert ones, and then forward reinforcement learning is employed to improve the similarity while considering safety concerns via a Lagrange multiplier. Besides, we provide a theoretical interpretation of LGAIL, which indicates that the proposed LGAIL can be guaranteed to learn a safe policy from unsafe expert data. At last, extensive experiments in OpenAI Safety Gym conclude the effectiveness of our approach.

1 INTRODUCTION

Imitation Learning (IL), which learns from expert data or expert policies to reproduce an expert policy, has achieved remarkable successes in various applications such as self-driving (Li et al., 2017; Pan et al., 2020), navigation (Hussein et al., 2018), and robot locomotion (Yuan & Kitani, 2020). Most of these algorithms are trained in simulated environments, in which agents are free to make mistakes. However, when deploying IL in real-world applications, the safety of agents is paramount (Amodei et al., 2016; Ray et al., 2019; Arora & Doshi, 2021). A policy that is trained without considering safety could generate improper or even harmful actions, and those actions may destroy the safety of agents, which must be avoided in safety-critical scenarios (Sinha et al., 2020).

Nevertheless, little attention has been paid to guarantee the safety of agents in IL. Zhang & Cho (2016) investigated the safety in behavioral cloning (BC) (Bain & Sammut, 1995), a branch of IL, and proposed an algorithm named SafeDAgger. Following SafeDAgger, Menda et al. (2019) advanced a modified version that uses the Gaussian Process (GP) (Rasmussen, 2003) to determine the confidence of whether the agent's decision is safe or not. However, the above-mentioned Safe IL approaches rely on access to an expert policy and demand that the expert policy is absolutely safe. Unfortunately, these two requirements–there should be an expert policy and the expert policy is absolutely safe–could hardly be satisfied in reality.

In contrast, we present a more practical Safe IL task. In the new task, we only require expert data rather than expert policies and the safety information provided by the environment. Besides, we do not assume that expert data are totally safe, *i.e.*, the expert data could contain a portion of unsafe data. We nominate this kind of data, which are not guaranteed to be purely safe, as "unsafe expert data" throughout this paper. Conducting Safe IL with unsafe expert data is more realistic because: (1) massive amounts of expert data such as online videos (Peng et al., 2018a) or MoCap data (Peng et al., 2018b) are available for IL, but it is hardly possible to require an expert who can always tell us the correct action during learning; (2) it is costly and laborious to obtain purely safe expert data because even experts could take dangerous actions (Council et al., 2003; Bickmore et al., 2018; Liu et al., 2020; Lattanzi & Freschi, 2021). In consequence, it is challenging to conduct IL from unsafe expert data to recover policies that can achieve expert-level performance and satisfy safety needs.

Unfortunately, to the best of our knowledge, the Safe IL task described above is of significance and needs to be solved urgently, but it has not been investigated until now.

To reproduce policies that can simultaneously achieve expert-level cumulative rewards and satisfy safety constraints by imitating unsafe expert data, we interpret this Safe IL task as a constrained optimization problem with Constrained Markov Decision Process (CMDP) (Altman, 1999), *i.e.*, the agent should try to behave as similarly as possible to the expert under safety constraints. Specifically, we introduce an auxiliary cost constraint to restrict the policy generated by Generative Adversarial Imitation Learning (GAIL) (Ho & Ermon, 2016), which leads to a constrained minimax saddle point problem. To tackle the difficult inequality constraint, we adopt a Lagrange multiplier technique to relax the constrained GAIL problem as an unconstrained one, abbreviated as LGAIL. Then, based on a stochastic variant of dual ascent algorithm, we propose a new two-stage optimization framework to solve LGAIL. Specifically, in the first stage, a discriminator is optimized to better measure the similarity between the agent-generated state-action pairs and the expert ones. In the second stage, forward reinforcement learning is employed to improve the similarity while considering safety concerns via a Lagrange multiplier. To the end, we summarize our contributions as three-fold:

- We formalize a new Safe IL task with CMDP, where the agent has access to several unsafe expert trajectories and the safety information provided by the environment.
- We develop a Safe IL algorithm–LGAIL, a neat yet effective way to tackle the new Safe IL task. We also provide a theoretical interpretation of LGAIL, which indicates the proposed LGAIL can be guaranteed to learn a safe policy from unsafe expert data.
- We carry out extensive experiments on various robot tasks in the OpenAI Safety Gym (Ray et al., 2019) to illustrate that LGAIL can work well in the novel Safe IL task defined in this paper and can serve as a baseline algorithm for future research.

2 Related Work

Safe Reinforcement Learning (Safe RL). RL with safety-critical constraints, also known as Safe RL, has received extensive attention in the past decades (Ray et al., 2019). The most popular way to deal with Safe RL is to convert it into a constrained optimization problem via CMDP (Altman, 1999). There are two major classes of methods to solve Safe RL featured by CMDP, *i.e.*, direct approaches and indirect approaches. Constrained Policy Optimization (CPO) (Achiam et al., 2017) is a representative algorithm of direct methods, in which the policy is optimized under the policy improvement and safety constraints. Yang et al. (2020) split the optimization problem in CPO into two steps: first, optimize the policy with consideration of only rewards; then project the optimized policy into the nearest safe policy. Two milestones of indirect algorithms are TRPO-Lagrangian and PPO-Lagrangian (Ray et al., 2019), which uses a Lagrange multiplier and shows outstanding performance of satisfying constraints. Stooke et al. (2020) improve the Lagrangian methods with PID control to reduce constraint-violating behaviors. However, the above methods cannot guarantee the safety of agents during training. To achieve the training safety of agents, another spectrum of Safe RL algorithms is developed based on Lyapunov functions (Chow et al., 2018; 2019).

Safe Imitation Learning (Safe IL). IL commits to reproduce an expert policy from expert data or expert policies. In general, IL can be divided into behavioral cloning (BC) (Bain & Sammut, 1995; Ross et al., 2011) and inverse reinforcement learning (IRL) (Abbeel & Ng, 2004). The major difference is that BC solves IL in a supervised learning manner, whereas IRL solves IL from the perspective of RL (Torabi et al., 2018). BC enjoys merits of simpleness and high efficiency but suffers from the compounding error and often fails to recover an expert policy compared to IRL (see Hussein et al. (2017) and its reference therein). It looks like that IL and batch reinforcement learning (Batch RL) (Lange et al., 2012; Fujimoto et al., 2019; Le et al., 2019) can solve the same problems. Actually, it is not exact because Batch RL and IL are dramatically different in terms of the data such that both domains cannot be compared. However, when it comes to Safe IL, there is few work. A representative method is SafeDAgger (Zhang & Cho, 2016), which is built on the BC framework DAGGER (Ross et al., 2011). SafeDAgger measures the difference between decisions of the learner and the expert while interacting with environments. When the difference goes beyond a predefined bound, the expert decision will be executed to ensure the safety of the learner. Menda et al. (2019) present EnsembleDAgger that uses an ensemble of neural networks to approximate the confidence

to determine whether it is safe to enforce the agent's decision. However, both algorithms require a safe expert policy, which is difficult to be satisfied in practice.

We consider a more practical task for Safe IL in this paper, where the agent is required to conduct Safe IL with unsafe expert data. Compared with Zhang & Cho (2016); Menda et al. (2019), there exist two main differences: first, there are no expert policies to teach the imitator; second, the provided expert data could be unsafe. These two differences dramatically increase the difficulty of conducting Safe IL. In addition, the considered Safe IL setting is different from IL from imperfect demonstration (Wu et al., 2019) that merely considers performance and neglects the safety issue, while our work simultaneously focuses on safety and performance issues.

3 PRELIMINARIES

Constrained Markov Decision Process (CMDP). CMDP (Altman, 1999; Achiam et al., 2017) is modeled by $(S, A, T, R, C, d_0, \gamma)$, where S is state space, A represents action space, $T = T(s'|s, a)$ is the environment transition dynamic, $R : S \times A \to \mathbb{R}$ is the reward function, $C : S \times A \to \mathbb{R}$ is the cost function, d_0 is the cost limit, and γ is the discount factor. Let $\pi(a_t|s_t) : S \times A \to [0, 1]$ be a stochastic policy for the agent. The cost in CMDP refers to safety. When we talk about "cost" in this paper, it indicates that we are focusing on safety. Let $J_R(\pi) = \mathbb{E}_{s_0,a_0,\dots}[R_0]$ with $R_t = \sum_{l=0}^{\infty} \gamma^l r_{t+l}$ denoting the expected discounted reward, where $s_0 \sim \rho_0(s_0)$, $a_t \sim \pi(a_t|s_t)$, $s_{t+1} \sim T(s_{t+1}|s_t, a_t)$, and $\rho_0(s_0)$ is the probability distribution of the initial state s_0. Similarly, the expected discounted cost is $J_C(\pi) = \mathbb{E}_{s_0,a_0,\dots}[C_0]$ with $C_t = \sum_{l=0}^{\infty} \gamma^l c_{t+l}$. The goal of Safe RL defined in Eq. (1) is to find the optimal policy $\pi^*(a_t|s_t)$ that simultaneously satisfies the cost limit,

$$\pi^* = \arg\max_{\pi} J_R(\pi) \quad s.t.\ J_C(\pi) \leq d_0. \tag{1}$$

Generative Adversarial Imitation Learning (GAIL). GAIL conducts IL by minimizing the divergence between experts' and agents' trajectories. The learned policy performs similarly to the expert when the trajectory sampled from the agent's policy matches that of the expert. It is formulated as the following minimax saddle point optimization (Guo et al., 2018; Shin & Kim, 2019a;b):

$$\min_{\theta} \max_{\omega} \ \mathbb{E}_{\pi_\theta}[\log D_w(s, a)] + \mathbb{E}_{\pi_E}[\log(1 - D_w(s, a))] - \beta\mathcal{H}(\pi_\theta), \tag{2}$$

where $D_w(s, a)$ is a discriminator that is parameterized with w, $\mathcal{H}(\pi) = \mathbb{E}_\pi[-\log \pi(a|s)]$ is the entropy of policy π, and $\beta \geq 0$ is a hyperparameter.

4 SAFE IMITATION LEARNING WITH A LAGRANGE MULTIPLIER

In this section, we present the proposed Safe IL paradigm, Lagrangian Generative Adversarial Imitation Learning (LGAIL). Below, we first formalize the new task of Safe IL with CMDP in Subsection 4.1. Then, the detailed description of LGAIL is presented in Subsections 4.2 and 4.3.

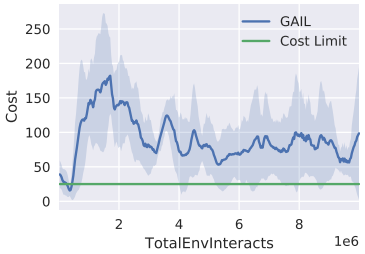


Figure 1: A toy example of IL from purely safe expert trajectories.

4.1 PROBLEM FORMULATION

Motivations. Three significant factors motivate us to study safe IL with unsafe expert data. First, it is natural that expert data may contain a portion of dangerous data due to the following two reasons: (1) even senior experts could not be immune to mistakes or dangerous decisions (Best, 1992; Culverhouse et al., 2003); (2) practical expert data often come from various sources with distinct qualities (Tangkaratt et al., 2020). As a result, it is likely that the expert data collected by sampling from varied experts include some unsafe actions or trajectories, which we define as "unsafe expert data". Second, ensuring the safety of agents is paramount in most applications. For example, in robot locomotion, a series of dangerous actions are likely to lead to the robot falling down, which may irrevocably damage the sophisticated robot (Yu et al., 2019). What's worse, in some safety-critical domains such as human-robot interaction, robots could cause human injuries if no special operations are designed for safety. Hence, it is of

significance to pay attention to safety issues in IL. Last but not least, we conduct a toy experiment to demonstrate that without considering the safety, it is hard to ensure the training safety and final safety. We employ the environment Safexp-PointGoal1-v0 in the OpenAI Safety Gym (Ray et al., 2019) and conduct ordinary GAIL with 30 safe expert trajectories whose Cost is 8.0 ± 8.3 and Return is 19.5 ± 2.8. Details on the environment and metrics such as Cost and Return are presented in Section 5. Every expert trajectory is safe such that its Cost is smaller than the cost limit $d_0 = 25$. From the IL results in Figure 1, surprisingly, even purely safe expert data are provided, the conventional IL algorithm–GAIL, is not able to reproduce a safe expert policy. In addition, the Cost goes beyond 150 at about 1.8 million interactions in Figure 1, which dramatically exceeds the cost limit. The high Cost during training means that traditional GAIL could not maintain the training safety as well. Therefore, with the three motivations and current problems of GAIL, we aim to solve the Safe IL task, which is formalized subsequently.

Compared to traditional IL in which safe expert trajectories are provided (Yang et al., 2019), the new task that considers safety during IL is different in that it does not require expert data to be absolutely safe. We formalize the fact that the expert data could be unsafe with an assumption.

Assumption 1 *We have access to a series of expert trajectories, in which some could be unsafe. We term this kind of expert data as "unsafe expert data".*

The unsafe expert trajectories are denoted as $\tau_E = \{\tau_E^1, \tau_E^2, ..., \tau_E^N\}$, and each trajectory τ_E^i where $i \in \{1, ..., N\}$ is composed of chronological states and actions. These expert trajectories can achieve high episodic cumulative rewards, but among them there are M expert trajectories that do not satisfy the safety constraints characterized by $J_C(\tau_E^j) \geq d_0, 1 \leq j \leq M$, where $0 \leq M \leq N$. If $M = 0$, there are no unsafe expert trajectories in τ_E, and τ_E is purely safe; if $M = N$, every trajectory in τ_E is unsafe, and τ_E is purely unsafe; if $0 < M < N$, τ_E is partially unsafe. Hence, the definition of "unsafe expert data" in the paper is quite universal, which covers the ideal situation that every expert trajectory is safe, a more practical situation that some expert data are unsafe, and the extreme situation that each expert trajectory is unsafe. Naively conducting IL with unsafe expert data will generate a policy that could be unsafe as well. To make it possible to achieve a safe agent, a reasonable assumption on the access to the safety information is made below.

Assumption 2 *The agent has direct access to the safety information from the environment.*

Assumption 2 makes sense in reality because safety functions are generally clear and straightforward to design compared to reward functions. For example, in autonomous driving, dangerous conditions such as collisions with pedestrians or cars can be easily identified (Shin & Kim, 2019a). Therefore, the aim is to obtain a safe policy utilizing unsafe expert data and the feedback of safety information from the environment. The task of interest of this paper is presented as follows:

The task of interest: Given unsafe expert trajectories τ_E in Assumption 1 and the safety information feedback in Assumption 2, we aim to find a policy that can mimic the expert as much as possible under given safety constraints.

The task is new compared to previous Safe IL research in which a safe expert policy is required (Zhang & Cho, 2016; Menda et al., 2019). Although conducting Safe IL in this task is arduous, it is worth investigating Safe IL due to its potential for practical applications compared to former tasks.

4.2 Safe Imitation Learning

In this new task of Safe IL, there are two learning objectives. The first one is that the agent should mimic the expert as much as possible via given expert trajectories when it comes to the episodic cumulative rewards. The second one is that the agent should behave safely to meet the safety constraints utilizing the environment feedback. The safety should be considered as a hard constraint because it represents physical requirement and should not be violated, which motivates us to model safe IL as constrained optimization, *i.e.*, the agent is supposed to mimic the expert as much as possible under safety constraints. Note that it is not a pure IL problem because the agent should behave unlike the expert in some states due to safety concerns. Thus, we formulate Safe IL on the top of GAIL as a constrained minimax saddle point optimization,

$$\min_{\theta} \max_{\omega} \mathbb{E}_{\pi_\theta}[\log D_w(s,a)] + \mathbb{E}_{\pi_E}[\log(1 - D_w(s,a))] - \beta \mathcal{H}(\pi_\theta) \; s.t. \; J_C(\pi_\theta) \leq d_0. \tag{3}$$

In other words, when optimizing the policy, similarities calculated by the discriminator D_w rewards as well as violations of safety constraints should be considered.

Algorithm 1 Lagrangian Generative Adversarial Imitation Learning (LGAIL)

Input: Expert trajectories τ_E, iteration number m, cost limit d_0, and learning rate η.
Parameter: Policy π_θ, discriminator D_w, and Lagrange multiplier λ.
for $i = 1$ **to** m **do**
  $\tau \sim \pi_\theta$ ▷ sample agent trajectories
  $\omega \leftarrow \arg\max_{\omega} \hat{\mathbb{E}}_\tau[\log D_w(s,a)] + \hat{\mathbb{E}}_{\tau_E}[\log(1 - D_w(s,a))]$ ▷ update discriminator D_w
  $\theta \leftarrow \arg\min_{\theta} \hat{\mathbb{E}}_\tau[\log D_w(s,a)] + \lambda(J_C(\pi_\theta) - d_0) - \beta\mathcal{H}(\pi_\theta)$ ▷ update policy π_θ
  $\lambda \leftarrow (\lambda + \eta(J_C(\pi_\theta) - d_0))_+$, where $(\cdot)_+ = \max\{0, \cdot\}$ ▷ update Lagrange multiplier λ
end for

4.3 Lagrangian Generative Adversarial Imitation Learning

To solve the Safe IL problem, we propose a two-stage optimization framework, LGAIL, whose pseudo-code is illustrated in Algorithm 1.

4.3.1 Imitation Learning with a Lagrange Multiplier

Directly solving the Safe IL task, which is a constrained optimization problem, is challenging. We employ a Lagrange multiplier to relax the constrained optimization problem into an unconstrained optimization one (Boyd et al., 2004), *i.e.*, the safety constraints are converted into penalties. As a result, we augment the policy improvement stage in GAIL with a Lagrange multiplier. Concretely, the constrained optimization problem in Eq. (3) can be solved by penalizing violations of safety constraints with a Lagrange multiplier while optimizing the policy to mimic the expert,

$$\min_{\theta} \max_{\omega} \mathbb{E}_{\pi_\theta}[\log D_w(s,a)] + \lambda[J_C(\pi_\theta) - d_0] - \beta\mathcal{H}(\pi_\theta) + \mathbb{E}_{\pi_E}[\log(1 - D_w(s,a))], \tag{4}$$

where λ is the Lagrange multiplier with $\lambda \geq 0$.

This optimizing target contains both rewards and costs, which can help imitate the expert as well as guarantee safety. There are two stages in LGAIL taking turns to (i) optimize a discriminator network to enhance its ability on judging the quality of state-action pairs, and (ii) improve the performance of the agent's policy with the discriminator network and safety feedback information enabled by a Lagrange multiplier. The Lagrange multiplier λ helps balance the competition between improving rewards and reducing costs, and it is dynamically updated according to $\lambda \leftarrow (\lambda + \eta(J_C(\pi_\theta) - d_0))_+$, where $(\cdot)_+ = \max\{0, \cdot\}$ and η is the learning rate. When the current policy is unsafe, λ will increase so that the penalty on violations of constraints will play a bigger role. On the contrary, λ would decrease so that the optimization concentrates more on mimicking the expert. LGAIL achieves: 1) the first objective by using GAIL to imitate the expert; 2) the second one with a Lagrange multiplier to force the agent to satisfy safety constraints, *i.e.*, the Lagrange multiplier will penalize the agent when the agent's behaviors are unsafe while imitating the unsafe expert.

4.3.2 Theoretical Interpretation

Safe IL defined in Eq. (3), is a constrained minimax saddle point problem. The inner max loop mainly optimizes ω to obtain a better discriminator $D_\omega(s,a)$, which assigns rewards to state-action pairs (s,a). The outer optimizing loop mainly optimizes the policy with consideration of both rewards and costs. Here, we only focus on the outer optimization loop. Assume the inner loop obtains an optimal ω^* as in Weng (2019), then the outer loop becomes,

$$\min_{\theta} \mathbb{E}_{\pi_\theta}[\log D_{\omega^*}(s,a)] - \beta\mathcal{H}(\pi_\theta) + \mathbb{E}_{\pi_E}[\log(1 - D_{\omega^*}(s,a))] \; s.t. \; J_C(\pi_\theta) \leq d_0. \tag{5}$$

The above constrained optimization problem can be solved with a Lagrange multiplier $\lambda \geq 0$. The Lagrangian function $L(\theta, \lambda, \omega^*)$ of LGAIL is written as follows,

$$\begin{aligned} L(\theta, \lambda, \omega^*) =& \mathbb{E}_{\pi_\theta}[\log D_{\omega^*}(s,a)] + \lambda[J_C(\pi_\theta) - d_0] \\ & - \beta\mathcal{H}(\pi_\theta) + \mathbb{E}_{\pi_E}[\log(1 - D_{\omega^*}(s,a))] + \delta_{\geq 0}(\lambda), \end{aligned} \tag{6}$$

where $\delta_{\geq 0}(\cdot)$ is an indicator function, *i.e.*, $\delta_{\geq 0}(\lambda) = 0$ if $\lambda \geq 0$, $\delta_{\geq 0}(\lambda) = \infty$ otherwise. The Lagrange multiplier $\lambda \geq 0$ penalizes violations of safety constraints to ensure that the policy is safe while optimizing the policy to mimic the expert. The outer loop optimization can be solved by dual ascent approaches. With dual ascent in a deterministic setting (Luo & Tseng, 1993; Andersson et al., 2016), it is guaranteed to arrive at a stationary point satisfying the first-order optimality conditions

$$0 = \nabla_\theta L(\theta, \lambda, w^*) \;\&\; 0 \in \nabla_\lambda L(\theta, \lambda, w^*). \tag{7}$$

Hence, it is clear from the above equation that $(J_C(\pi_\theta) - d_0) \in \mathcal{N}_{\geq 0}(\lambda)$, where $\mathcal{N}_{\geq 0}(x)$ is the normal cone of set $\{x | x \geq 0\}$. Consequently, we obtain $J_C(\pi_\theta) \leq d_0$, meaning that LGAIL is guaranteed to find a policy that satisfies the safety criterion with deterministic gradient descent.

In practical implementations, we use stochastic gradient descent with samples by interacting with the environment. Besides, we do not exactly solve the inner and outer optimization problems. Instead, we employ the approximation solutions of the inner and outer loops, and alternatively optimize them. As a result, these approximation methods may cause instability on the convergence curve of LGAIL. But our experiments empirically demonstrate that LGAIL finally finds a safe policy.

4.3.3 Practical Implementation

Eq. (4) defines the optimization target for LGAIL, in which three parameters are involved (θ for policy, w for discriminator, and Lagrage multiplier λ). We denote the optimization target as $L(\theta, \lambda, w)$. Since we cannot access the expert policy, expert and agent trajectories, τ_E and τ, are used to approximate the loss. Hence, we use $\hat{\mathbb{E}}$ to represent an approximation of ideal expectation with sampled data. The update law for the discriminator is,

$$\nabla_w L(\theta, \lambda, w) = \hat{\mathbb{E}}_\tau[\nabla_w \log D_w(s, a)] + \hat{\mathbb{E}}_{\tau_E}[\nabla_w \log(1 - D_w(s, a))]. \tag{8}$$

For the optimization of the policy, the rapid variation of λ may affect the training stability. Hence, we adopt a technique (Stooke et al., 2020) to regulate the policy gradient

$$\nabla_\theta L(\theta, \lambda, w) = \hat{\mathbb{E}}_\tau[\nabla_\theta \log \pi_\theta(a|s) \frac{1}{1+\lambda}(Q^r(s, a) - \lambda Q^c(s, a))] - \beta \nabla_\theta H(\pi_\theta), \tag{9}$$

in which $Q^r(\bar{s}, \bar{a}) = \hat{\mathbb{E}}_\tau[-\log(D_w(s, a))|s_0 = \bar{s}, a_0 = \bar{a}]$ and $Q^c(s, a) = \hat{\mathbb{E}}_\tau[C(s, a))|s_0 = \bar{s}, a_0 = \bar{a}]$. Two separate neural networks are adopted to maintain accurate approximation of q-values for reward and cost. Besides, during training, the Lagrange multiplier λ is dynamically updated to ensure the agent satisfy safety constraints according to,

$$\nabla_\lambda L(\theta, \lambda, w) = \hat{\mathbb{E}}_\tau(J_C(\pi_\theta) - d_0), \tag{10}$$

in which the agent trajectories τ are used to estimate $J_C(\pi_\theta)$.

5 Experiments

We investigate whether our algorithm LGAIL is able to solve the new Safe IL task in this paper, *i.e.*, whether LGAIL has the ability to reproduce safe expert behaviors from unsafe expert data with the safety information feedback from the environment. We introduce our experiments from three aspects, setups (Subsection 5.1), results (Subsection 5.2), and discussions (Subsection 5.3).

5.1 Setups

In the experiments, we adopt six standard Safety Gym environments (Ray et al., 2019) to demonstrate the ability of LGAIL. In terms of robots, we use Point, Car, and Doggo; in terms of tasks, Goal and Button are employed. The level of difficulty of the employed environments is set to 1. More details on environments and unsafe expert data are in Appendix A.

Baselines. The Safe IL task in this paper is constructed for the first time, so there are no corresponding baselines to compare. We select one representative IL algorithm, GAIL (Ho & Ermon, 2016), to serve as the baseline. Unfortunately, we are not able to compare with SafeDAgger (Zhang & Cho, 2016) or EnsembleDAgger (Menda et al., 2019) because there are no expert policies in our

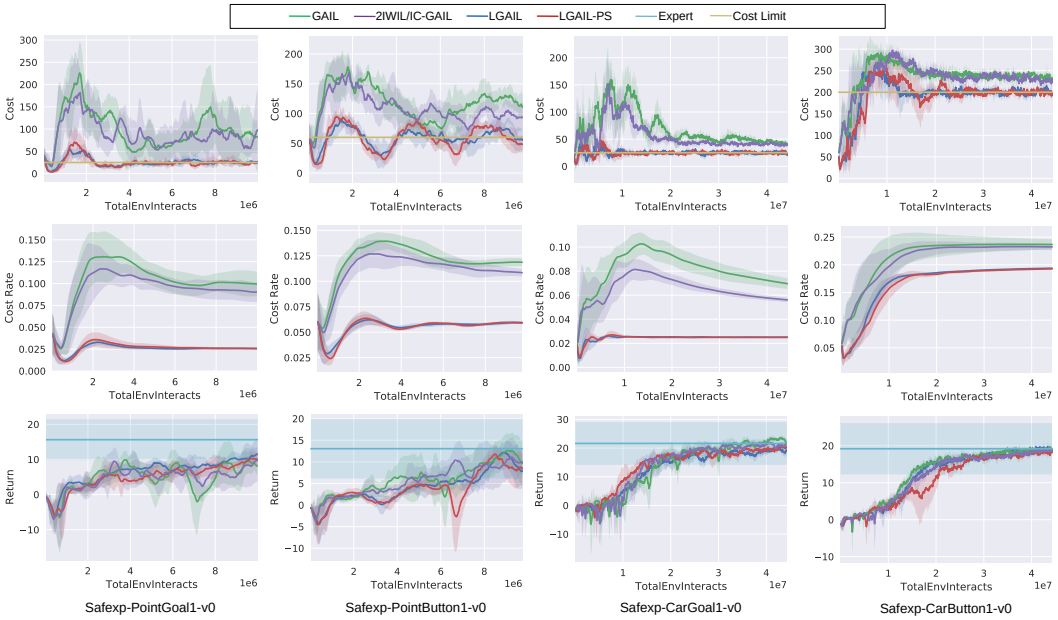


Figure 2: Learning curves of LGAIL and the other baselines on Safety Gym benchmarks. Performance is measured with Cost, Cost Rate, and Return. The x-axis represents time steps of interactions with the environment. Each algorithm is evaluated with 5 random seeds.

Table 1: Summary of quantitative results. The columns represent the algorithms, while the rows represent environments and metrics. Each result is averaged over 30 trails of a policy.

Environment		LGAIL	LGAIL-PS	GAIL	2IWIL/IC-GAIL
Safexp-PointGoal1-v0	Cost	24.7±4.3	**22.1±3.2**	81.8±35.5	103.0±98.2
	Cost Rate	**0.026**	**0.026**	0.099	0.09
	Return	**11.5±2.1**	10.1±2.1	8.2±2.6	9.8±6.0
Safexp-PointButton1-v0	Cost	56.3±22.0	**52.4±18.1**	109.6±28.2	91.2±14.0
	Cost Rate	**0.059**	**0.059**	0.119	0.108
	Return	7.6±3.7	8.2±3.9	9.2±5.4	**9.7±2.9**
Safexp-CarGoal1-v0	Cost	24.7±2.2	**21.6±2.1**	41.6±7.8	43.9±10.6
	Cost Rate	**0.025**	**0.025**	0.07	0.056
	Return	18.9±2.3	19.8±1.1	20.7±2.0	**21.4±1.1**
Safexp-CarButton1-v0	Cost	**192.8±15.5**	195.1±19.1	232.4±24.3	251.0±23.0
	Cost Rate	0.194	**0.193**	0.237	0.232
	Return	19.1±1.1	18.0±1.1	**19.2±1.2**	18.2±1.5

task. We also could not compare with Le et al. (2019) because Batch RL needs data with rewards and is not allowed to interact with the environment, whereas in our IL task the unsafe expert data do not contain rewards. In addition, we relax the exact problem formulation of LGAIL to compare with IL algorithms of learning from imperfect data (2IWIL and IC-GAIL) (Wu et al., 2019). More details on 2IWIL and IC-GAIL are in Appendix B. Besides, we conduct LGAIL with purely safe expert data and denote it as LGAIL-PS (LGAIL from Purely Safe expert data).

Metrics. To comprehensively measure the performance of all algorithms, three metrics are employed, *i.e.*, Cost, Cost Rate, and Return. Cost $J_C(\pi_\theta)$ is the average episodic sum of costs, while Return $J_R(\pi_\theta)$ is the average episodic return. Cost Rate is the rate that can be obtained by dividing the total sum of costs of the whole training process by the total number of agent-environment interactions. Cost and Cost Rate are related to the safety of the agent: the smaller they are, the safer the agent is considered. Although both metrics are related to safety, Cost focuses on measuring the current safety of a policy, while Cost Rate emphasizes the safety of whole training process. Hence,

Cost Rate could be interpreted as a metric for the training safety to some extent. Return is used to evaluate the performance of mimicking the expert.

5.2 RESULTS

In this subsection, we present the experiment results of the proposed algorithm–LGAIL. Learning curves of four environments are presented in Figure 2, while quantitative results are in Table 1. More experiment results are deferred to Appendix C. From Figure 2 and Table 1, it is clear that LGAIL (including LGAIL and LGAIL-PS) is able to reproduce a safe policy that can satisfy the safety constraints with comparable performance in imitating the expert.

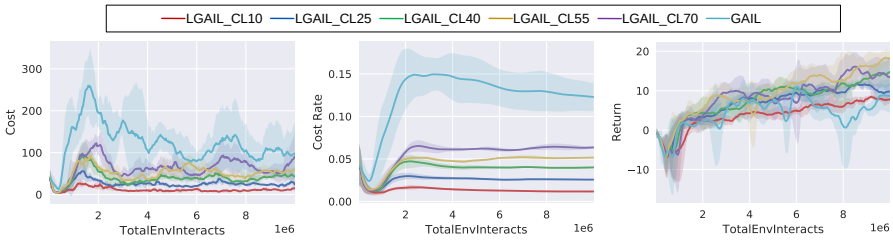


Figure 3: Impact of the cost limit on LGAIL with purely unsafe expert data. In the legend LGAIL_CL$\{x\}$, x represents the cost limit d_0.

Safety. It can be seen that LGAIL can achieve much lower Cost and lower Cost Rate, which means that LGAIL is safer compared to other baselines and the experts. For example, in Safexp-PointGoal1-v0, the peak values of Cost for LGAIL and GAIL are about 60 and 230, respectively. Besides, the Cost Rate of LGAIL is about only a quarter of GAIL's Cost Rate, meaning that the violations of constraints using GAIL have been reduced by three quarters with LGAIL. In particular, LGAIL is able to drive the learning process to generate an agent that can satisfy the given constraint threshold. The lower Cost and lower Cost Rate mean that the safety has both been improved during training and at the end of training. Specifically, LGAIL-PS performs similarly to LGAIL, which means that our algorithm is robust to the type of the expert data no matter it is purely safe or not. Compared to GAIL, 2IWIL/IC-GAIL slightly improves the safety measured by Cost and Cost Rate because 2IWIL/IC-GAIL tries to learn from purely safe expert data. These results verify that a portion of unsafe expert data could cause a negative impact on the safety of IL algorithms. Although 2IWIL/IC-GAIL performs slightly superior to GAIL in terms of safety, the performance of 2IWIL/IC-GAIL is still far from satisfactory, indicating that GAIL cannot recover a safe policy from purely safe expert data. We provide further discussions on this phenomenon in the next subsection.

Return. We observe that all the algorithms can achieve the same level of performance. In other words, with little sacrifice in Return, LGAIL obtains a notably safer agent compared against other baselines. In Doggo tasks, LGAIL performs slightly worse than the other baselines. We think that there are two possible reasons: (1) to keep safe, the agent in LGAIL should try to avoid and to keep away from dangerous areas. This means that the agent should travel the long way around. As a result, the rewards that LGAIL achieves in fixed steps would decrease. On the contrary, GAIL and 2IWIL/IC-GAIL do not take safety into consideration, so they can walk across dangerous areas to achieve higher rewards; (2) LGAIL seeks a balance between rewards and costs, and adopts a more conservative exploration strategy, leading to marginal performance degradation. When a policy is unsafe, the Lagrange multiplier will increase and penalize the policy to ensure safety. Therefore, LGAIL would take actions that are more conservative when it explores in the environment. In complex environments, exploration is important for discovering better policies. Although LGAIL might perform marginally worse than GAIL in complex environments regarding Return, the safety of the agent of LGAIL has been enhanced dramatically, which is paramount in safety-critical environments when deploying IL algorithms.

Furthermore, we test the extreme case where expert data are purely unsafe in environment Safexp-PointGoal1-v0. We employ 15 purely unsafe expert trajectories, with their Cost 69.5 ± 15.3 and Return 18.1 ± 2.4. In particular, we adjust the cost limit d_0 from 10 to 70 to investigate its impact on safety and reward performance, whose results are shown in Figure 3. From the perspective of Cost and Cost Rate, it is clear that LGAIL is able to obtain a safe agent that satisfies the cost constraint

with purely unsafe expert data, whereas traditional GAIL cannot. Namely, given a fixed d_0 no matter it is large or small before training, LGAIL is able to reproduce a policy such that $J_C(\pi_\theta) \leq d_0$. With the decrease in the cost limit, the performance of the agent after training decreases slightly. Even if the safety of the agent in LGAIL has been improved dramatically, the performance of LGAIL is comparable to that of GAIL.

In summary, the two-stage optimization framework, LGAIL, is able to reproduce a safe policy with unsafe expert data and the safety information feedback.

5.3 Discussions

It is worthy of investigating why GAIL could not reproduce safe policies with purely safe expert data. We conduct experiments to test the impact of the amount of expert data and the diversity of expert data on the safety performance of GAIL. Some results are shown in Figure 4, while experiment details and more results are deferred to Appendix C. From Figure 4, it is clear that: (1) GAIL usually fails to recover a safe policy even with abundant purely safe expert data; (2) increasing the number of expert data does not help improve the performance of GAIL in terms of both rewards and costs; (3) GAIL performs worse with expert data that are sampled from multiple experts.

In our opinion, there are three possible reasons: (1) The purely safe expert data are unbalanced. We think that the expert data contain more information on how to achieve rewards compared to the information on how to be safe. Every safe expert trajectory achieves high rewards but low costs, which means that rewards are dense while costs are sparse. As a result, GAIL is likely to mainly develop the ability to accomplish tasks but neglecting the connotative ability to be safe. (2) GAIL could not adapt well to dynamic environments due to the poor generalization ability. GAIL employs the RL algorithms to serve as the generator, and RL algorithms often struggle with generalization problems. Hence, GAIL is likely to generalize poorly in dynamic environments such that the recovered policy could be unsafe. (3) Expert data sampled from a mixture of expert policies could provide opposite information about safety. Different experts have their own preferences, which may mislead the agent to dangerous actions. In contrast, our algorithm LGAIL explicitly considers safety issues during imitating and regards them as constraints to regulate the IL process. This explicit modeling enables LGAIL to generate policies with guaranteed safety.

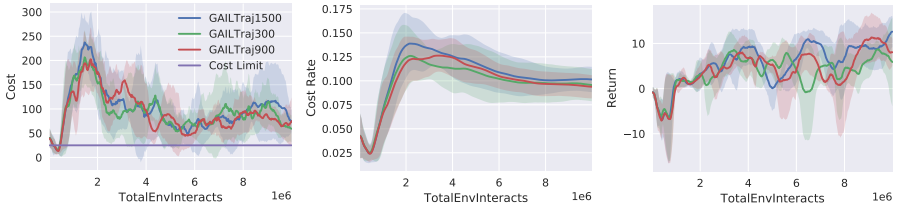


Figure 4: Impact of the number of expert data on GAIL with purely safe expert data. In the legend GAILTraj$\{x\}$, x represents the number of expert trajectories.

6 Conclusion

In this paper, a new but more practical Safe IL task is constructed, in which an agent has access to the safety information directly via interacting with the environment and several unsafe expert trajectories. To conduct Safe IL in this task, we develop a two-stage optimization framework, dubbed LGAIL, which can successfully imitate the expert and produce safety guaranteed policies. LGAIL treats the Safe IL task as a constrained optimization problem, in which the agent tries to maximize the cumulative episodic reward under safety constraints. LGAIL turns the constrained optimization problem into a corresponding unconstrained one with a Lagrange multiplier. The effectiveness and performance are illustrated and validated in extensive OpenAI Safety Gym benchmarks, meaning that our algorithm is able to deal with the new Safe IL task. In addition, the safety of agents during training is also enhanced dramatically compared to the baselines. Although the training safety of LGAIL is significantly enhanced, LGAIL fails to strictly maintain the safety of agents during training. A promising future direction would be achieving the training safety in Safe IL.

REPRODUCIBILITY STATEMENT

We acknowledge the importance of reproducibility for research work and try whatever we can to ensure the reproducibility of our work. We first introduce the environments used in detail in Appendix A. Since we are investigating a new safe imitation learning task, there are no existing data to conduct experiments. Hence, we present how we obtain expert data for this new task in Appendix A. As for the implementation of our algorithm, details such as hyperparameters are provided in Appendix B. Finally, we introduce error bars as well as the computing resources in Appendix C. Our codes and data will be released upon publication.

REFERENCES

Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 1, 2004.

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. *arXiv preprint arXiv:1705.10528*, 2017.

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

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Fredrik Andersson, Marcus Carlsson, and Carl Olsson. Convergence of dual ascent in non-convex/non-differentiable optimization. *arXiv preprint arXiv:1609.06576*, 2016.

Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, pp. 103500, 2021.

M. Bain and C. Sammut. A framework for behavioural cloning. *In Machine Intelligence*, 15:103–129, 1995.

Chris F Best. Even experts make mistakes. *Risk Management*, 39(1):48–50, 1992.

Timothy W Bickmore, Ha Trinh, Stefan Olafsson, Teresa K O'Leary, Reza Asadi, Nathaniel M Rickles, and Ricardo Cruz. Patient and consumer safety risks when using conversational assistants for medical information: an observational study of siri, alexa, and google assistant. *Journal of medical Internet research*, 20(9):e11510, 2018.

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

Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. In *Advances in neural information processing systems*, pp. 8092–8101, 2018.

Yinlam Chow, Ofir Nachum, Aleksandra Faust, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. Lyapunov-based safe policy optimization for continuous control. *arXiv preprint arXiv:1901.10031*, 2019.

Forrest M Council, David L Harkey, Daniel T Nabors, Asad J Khattak, and Yusuf M Mohamedshah. Examination of fault, unsafe driving acts, and total harm in car-truck collisions. *Transportation research record*, 1830(1):63–71, 2003.

Phil F Culverhouse, Robert Williams, Beatriz Reguera, Vincent Herry, and Sonsoles González-Gil. Do experts make mistakes? a comparison of human and machine indentification of dinoflagellates. *Marine ecology progress series*, 247:17–25, 2003.

Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. Openai baselines. https://github.com/openai/baselines, 2017.

Scott Fujimoto, Edoardo Conti, Mohammad Ghavamzadeh, and Joelle Pineau. Benchmarking batch deep reinforcement learning algorithms. *arXiv preprint arXiv:1910.01708*, 2019.

Yijie Guo, Junhyuk Oh, Satinder Singh, and Honglak Lee. Generative adversarial self-imitation learning. *arXiv preprint arXiv:1812.00950*, 2018.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in neural information processing systems*, pp. 4565–4573, 2016.

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

Ahmed Hussein, Eyad Elyan, Mohamed Medhat Gaber, and Chrisina Jayne. Deep imitation learning for 3d navigation tasks. *Neural computing and applications*, 29(7):389–404, 2018.

Sascha Lange, Thomas Gabel, and Martin Riedmiller. Batch reinforcement learning. In *Reinforcement learning*, pp. 45–73. Springer, 2012.

Emanuele Lattanzi and Valerio Freschi. Machine learning techniques to identify unsafe driving behavior by means of in-vehicle sensor data. *Expert Systems with Applications*, 176:114818, 2021.

Hoang Le, Cameron Voloshin, and Yisong Yue. Batch policy learning under constraints. In *International Conference on Machine Learning*, pp. 3703–3712. PMLR, 2019.

Yunzhu Li, Jiaming Song, and Stefano Ermon. Infogail: Interpretable imitation learning from visual demonstrations. In *Advances in Neural Information Processing Systems*, pp. 3812–3822, 2017.

Zuxin Liu, Hongyi Zhou, Baiming Chen, Sicheng Zhong, Martial Hebert, and Ding Zhao. Safe model-based reinforcement learning with robust cross-entropy method. *arXiv preprint arXiv:2010.07968*, 2020.

Zhi-Quan Luo and Paul Tseng. On the convergence rate of dual ascent methods for linearly constrained convex minimization. *Mathematics of Operations Research*, 18(4):846–867, 1993.

Kunal Menda, Katherine Driggs-Campbell, and Mykel J. Kochenderfer. Ensembledagger: A bayesian approach to safe imitation learning. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5041–5048, 2019. doi: 10.1109/IROS40897.2019.8968287.

Yunpeng Pan, Ching-An Cheng, Kamil Saigol, Keuntaek Lee, Xinyan Yan, Evangelos A Theodorou, and Byron Boots. Imitation learning for agile autonomous driving. *The International Journal of Robotics Research*, 39(2-3):286–302, 2020.

Xue Bin Peng, Pieter Abbeel, Sergey Levine, and Michiel van de Panne. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Transactions on Graphics (TOG)*, 37(4):1–14, 2018a.

Xue Bin Peng, Angjoo Kanazawa, Jitendra Malik, Pieter Abbeel, and Sergey Levine. Sfv: Reinforcement learning of physical skills from videos. *ACM Transactions on Graphics (TOG)*, 37(6):1–14, 2018b.

Carl Edward Rasmussen. Gaussian processes in machine learning. In *Summer school on machine learning*, pp. 63–71. Springer, 2003.

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

S. Ross, G. Gordon, and D. 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, 2011.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897, 2015.

MyungJae Shin and Joongheon Kim. Adversarial imitation learning via random search. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2019a.

MyungJae Shin and Joongheon Kim. Randomized adversarial imitation learning for autonomous driving. *arXiv preprint arXiv:1905.05637*, 2019b.

Aman Sinha, Matthew O'Kelly, Russ Tedrake, and John C Duchi. Neural bridge sampling for evaluating safety-critical autonomous systems. *Advances in Neural Information Processing Systems*, 33, 2020.

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by pid lagrangian methods. *arXiv preprint arXiv:2007.03964*, 2020.

Voot Tangkaratt, Bo Han, Mohammad Emtiyaz Khan, and Masashi Sugiyama. Variational imitation learning with diverse-quality demonstrations. In *International Conference on Machine Learning*, pp. 9407–9417. PMLR, 2020.

Faraz Torabi, Garrett Warnell, and Peter Stone. Generative adversarial imitation from observation. *arXiv preprint arXiv:1807.06158*, 2018.

Lilian Weng. From gan to wgan. *arXiv preprint arXiv:1904.08994*, 2019.

Yueh-Hua Wu, Nontawat Charoenphakdee, Han Bao, Voot Tangkaratt, and Masashi Sugiyama. Imitation learning from imperfect demonstration. *arXiv preprint arXiv:1901.09387*, 2019.

Chao Yang, Xiaojian Ma, Wenbing Huang, Fuchun Sun, Huaping Liu, Junzhou Huang, and Chuang Gan. Imitation learning from observations by minimizing inverse dynamics disagreement. In *Advances in Neural Information Processing Systems*, pp. 239–249, 2019.

Tsung-Yen Yang, Justinian Rosca, Karthik Narasimhan, and Peter J Ramadge. Projection-based constrained policy optimization. In *ICLR*, 2020.

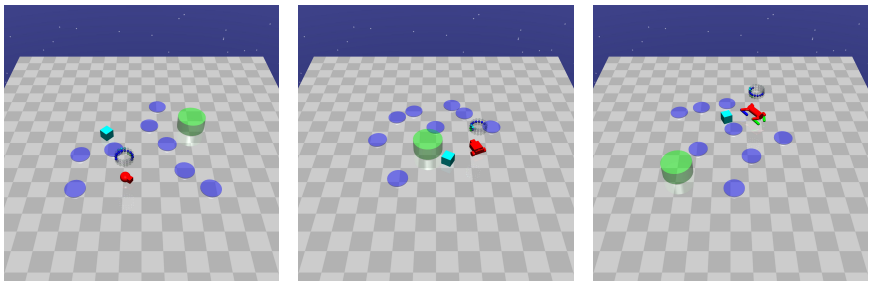
Ming Yu, Zhuoran Yang, Mladen Kolar, and Zhaoran Wang. Convergent policy optimization for safe reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 3127–3139, 2019.

Ye Yuan and Kris Kitani. Residual force control for agile human behavior imitation and extended motion synthesis. *arXiv preprint arXiv:2006.07364*, 2020.

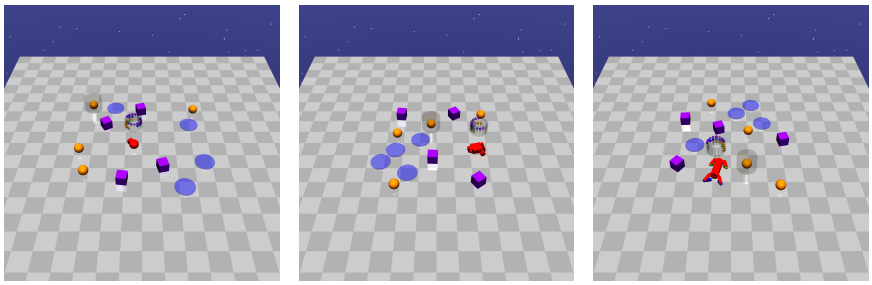
Jiakai Zhang and Kyunghyun Cho. Query-efficient imitation learning for end-to-end autonomous driving. *arXiv preprint arXiv:1605.06450*, 2016.

A ENVIRONMENT AND EXPERT DATA

In this section, we introduce the OpenAI Safety Gym benchmarks (Ray et al., 2019) used in our experiments and give details on how to generate unsafe expert data.



(a) Safexp-PointGoal1-v0 (b) Safexp-CarGoal1-v0 (c) Safexp-DoggoGoal1-v0



(d) Safexp-PointButton1-v0 (e) Safexp-CarButton1-v0 (f) Safexp-DoggoButton1-v0

Figure 5: Screenshots of the OpenAI Safety Gym environments. In Safexp-PointGoal1-v0, the red Point should navigate to the green cylinder while avoiding the purple hazards on the floor.

A.1 ENVIRONMENT OVERVIEW

OpenAI Safety Gym (Ray et al., 2019) is a highly configurable environment, which supports users to construct desired environments with different robots, tasks, constraints, and observation spaces. In general, tasks in Safety Gym demand the robot to navigate dangerous environments including hazards and vases. There are three optional robots, *i.e.*, Point, Car, and Doggo, while three task options are offered, *i.e.*, Goal, Button, and Push. Constraints such as hazards and vases can be selected and placed into the environment. The information that an agent could receive may come from standard robot sensors, velocity sensors, and lidars. Furthermore, the level of difficulty of the man-made environment can be adjusted by increasing or decreasing the number of constraints. Generally, Safety Gym is such a huge system that it cannot be explained in detail due to the various configurable choices available.

Therefore, to give an intuitive understanding of the environments, we introduce a standard Safety Gym environment–Safexp-PointGoal1-v0, which is shown in Figure 5. As can be interpreted from its name, the robot in this environment is Point (the red object in Figure 5(a)), a small robot with two actuators, one for turning and the other for moving forward/backward; the task is Goal, which means that the robot should move to a goal position as depicted by the green area in Figure 5(a); the number "1" after the task Goal represents the difficulty level of this task. In terms of constraints, there are several hazards (purple circles on the floor in Figure 5(a)) that are randomly placed during the environment initialization. When the robot steps into a hazardous area, the cost indicator c_t will be 1; otherwise, $c_t = 0$ at each step. One episode will end after 1,000 steps. During the 1,000 steps, if the goal has been achieved, a new goal will be randomly placed on the map. For more details on the Safety Gym, we refer the readers to Ray et al. (2019).

A.2 Environment Specifications

The specifications of the tested environments are listed in Table 2.

Table 2: Specifications of the OpenAI Safety Gym Benchmarks.

Environment	State Space	Action Space	Max-Step
Safexp-PointGoal1-v0	60	2	1000
Safexp-PointButton1-v0	76	2	1000
Safexp-CarGoal1-v0	72	2	1000
Safexp-CarButton1-v0	88	2	1000
Safexp-DoggoGoal1-v0	104	12	1000
Safexp-DoggoButton1-v0	120	12	1000

A.3 Unsafe Expert Data

As stressed throughout the paper, "unsafe expert data" are provided to enable the Safe IL. We want to emphasize again that our "unsafe expert data" are composed of safe expert trajectories and a portion of unsafe expert data. This kind of expert data is of practical significance because collecting purely safe expert data is costly and arduous. Here, we demonstrate how to generate such expert data. First, we use the Safe RL algorithm TRPO-Lagrangian implementation (Trust Region Policy Optimization Lagrangian) in Ray et al. (2019) to train an agent with a given cost limit. After training, an agent, which can achieve high cumulative rewards and satisfy the cost limit, is obtained. This agent can be regarded as a safe expert, and we can get a series of expert data by executing this policy in the Safety Gym environments. Although this agent is considered to be safe in most cases, some trajectories sampled from it could be unsafe due to dynamically changing environments. This means that safe experts may still make mistakes and take dangerous actions, which is consistent with Assumption 1.

As a result, we can sample both safe trajectories and unsafe trajectories using such a safe expert. With consideration of the fact that practical expert data may come from a variety of sources, we also generate the data with multiple expert policies. In particular, for every Safety Gym environment, we use TRPO-Lagrangian to train three safe experts from scratch separately. Default hyper-parameters in Ray et al. (2019) are adopted and the cost limit for each environment are listed in Table 3. After training, we construct 10 safe expert trajectories and 5 unsafe expert trajectories by sampling from each expert. Both states and actions of the expert are recorded sequentially and a trajectory contains $1,000$ states and actions. Since there are three experts, we obtain a total number of 45 expert trajectories for each environment, in which 30 trajectories are safe and the other 15 trajectories are unsafe. The 45 expert trajectories are what we defined as "unsafe expert data", and no labels are provided to indicate whether one expert trajectory is safe or not during imitating.

Table 3: Cost limits for training safe experts.

Environment	Cost Limit d_0
Safexp-PointGoal1-v0	25
Safexp-PointButton1-v0	60
Safexp-CarGoal1-v0	25
Safexp-CarButton1-v0	200
Safexp-DoggoGoal1-v0	60
Safexp-DoggoButton1-v0	250

B Implementation Details

We implement LGAIL based on two open source codes, OpenAI Baselines (Dhariwal et al., 2017) and Safety Starter Agents (Ray et al., 2019). Following Dhariwal et al. (2017), we use the RL algorithm Trust Region Policy Optimization (TRPO) (Schulman et al., 2015) to serve as the generator. We adopt the discriminator from OpenAI Baselines to replace the reward that is fed back from environments in Safety Starter Agents. We present the number of expert trajectories and the complete hyper-parameters used for imitation learning in Table 4 and Table 5.

We want to discuss a little bit more about one of the baselines, 2IWIL/IC-GAIL (Wu et al., 2019), *i.e.*, algorithms of learning from imperfect demonstration. The basic problem for learning from imperfect demonstration is that expert data could be sampled from experts with different qualities (Wu et al., 2019). Note that the quality here stands for the performance of the expert. In other words, some expert data are sampled from optimal policies while others are sampled from sub-optimal policies. The expert data sampled from sub-optimal policies could mislead the imitator to sub-optimal performance. Besides, only a small portion of expert data is labeled with confidence scores. If the confidence score $conf(s, a) = 1$, then the state-action pair (s, a) is sampled from optimal policies. On the contrary, $conf(s, a) = 0$ means that (s, a) is sampled from sub-optimal policies. Essentially, the aim of their solutions, 2IWIL and IC-GAIL, is to find all the state-action pairs that are sampled from optimal policies and learn from these optimal data without distractions of sub-optimal data (Wu et al., 2019). Therefore, in our experiments, we conduct imitation learning from purely safe expert data, which is the ultimate form of 2IWIL and IC-GAIL.

Table 4: Number of expert trajectories. We abbreviate trajectories as Trajs.

Environment	Total Expert Trajs	Safe Expert Trajs	Unsafe Expert Trajs
Safexp-PointGoal1-v0	45	30	15
Safexp-PointButton1-v0	45	30	15
Safexp-CarGoal1-v0	45	30	15
Safexp-CarButton1-v0	45	30	15
Safexp-DoggoGoal1-v0	45	30	15
Safexp-DoggoButton1-v0	45	30	15

Table 5: Hyper-parameters in experiments.

Hyper-parameters	Value
Common parameters	
Network size (Except the discriminator network)	(256,256)
Network size (Discriminator network)	(100,100)
Activation	$tanh$
Batch size	3,000
Optimizer	Adam
Generator network update times	1
Discriminator network update times	1
Common parameters for TRPO	
Generalized Advantage Estimation Gamma	0.99
Generalized Advantage Estimation Lambda	0.97
Maximum KL	0.01
Learning rate (Value network)	1×10^{-3}
Value iteration	80
Policy entropy	0.0
Discriminator parameters	
Learning rate (Discriminator network)	3×10^{-4}
Discriminator entropy	1×10^{-3}
Penalty parameters	
Initial penalty	1
Penalty learning rate	5×10^{-2}

C Additional Experiments

We present more experimental results (including the quantitative results) in different environments with various configurations here to further validate the proposed algorithm–LGAIL.

C.1 COMPUTING RESOURCES

We use CPUs to run our experiments. The model name of the CPU is Intel(R) Xeon(R) Gold 6146 CPU @ 3.20GHz. The computation time for each environment is provided in Table 6.

Table 6: Computation time.

Environment	Time
Safexp-PointGoal1-v0	about 4 hours
Safexp-PointButton1-v0	about 4 hours
Safexp-CarGoal1-v0	about 13 hours
Safexp-CarButton1-v0	about 20 hours
Safexp-DoggoGoal1-v0	about 14 hours
Safexp-DoggoButton1-v0	about 20 hours

C.2 EXPERIMENTS ON DOGGO TASKS

The learning curves in Safexp-DoggoGoal1-v0 and Safexp-DoggoButton1-v0 are presented in Figure 6, and quantitative results of these two environments are listed in Table 7. We only conduct LGAIL with unsafe expert data rather than purely safe expert data because the former is more complex, *i.e.*, we do not conduct experiments of LGAIL-PS. Even in these complex environments, the proposed algorithm LGAIL can still mimic the expert under safety constraints.

The phenomenon that LGAIL performs slightly worse than the other baselines has been discussed in the paper. For Safexp-DoggoButton1-v0, the phenomenon that LGAIL did not reduce the cost is because the Cost of LGAIL is lower than the cost limit $d_0 = 250$. According to our algorithm, the Lagrange multiplier will be zero if the current policy satisfies the cost limit. In other words, LGAIL focuses on improving the rewards when the policy is safe. Hence, the learning curve of LGAIL in Safexp-DoggoButton1-v0 is reasonable. To demonstrate that LGAIL is able to reduce costs, we also conduct new experiments with lower cost limit $d_0 = 200$. The learning curves of LGAIL in Safexp-DoggoButton1-v0 with cost limit $d_0 = 200$ are presented in Figure 7.

C.3 EXPERT PERFORMANCE

The performance of the expert data is presented in Table 8. As we discussed above, we sample 10 trajectories from each expert. Besides, during sampling, we select trajectories according to specific reward or safety desires. The performance of the expert data in Table 8 is calculated from the sampled data. However, Safety Gym is a dynamically changing environment such that it is not enough to evaluate an expert with only 10 trajectories. Hence, we also provide the performance of the expert in Table 9, which is evaluated with 100 trajectories. As we can see from Table 9, the variance of an expert is relatively high. So even we test the expert with 100 trajectories, the performance of the expert could vary if we retest it. In the learning curves, we plot the expert performance rather than the performance of expert data because the former is fairer.

Table 7: Summary of quantitative results. The columns represent the algorithms, while the rows represent environments and metrics. Each result is averaged over 30 trails of a policy.

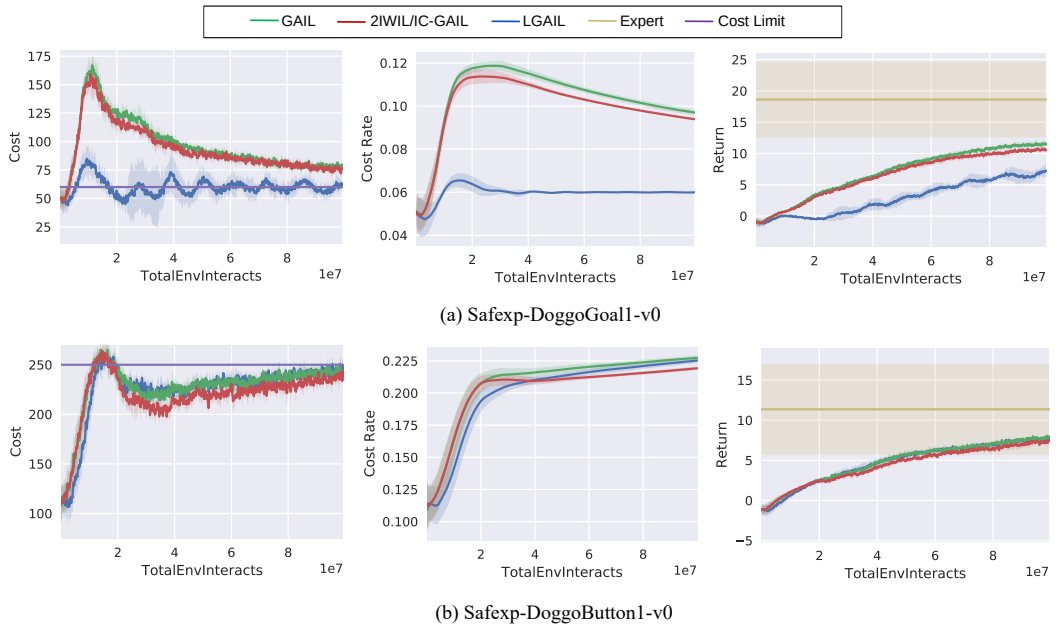
Environment		LGAIL	LGAIL-PS	GAIL	2IWIL/IC-GAIL
Safexp-DoggoGoal1-v0	Cost	58.9±7.6	-	80.4±7.5	73.1±6.8
	Cost Rate	0.06	-	0.097	0.094
	Return	7.2±1.1	-	11.6±0.5	10.5±0.3
Safexp-DoggoButton1-v0	Cost	241.7±16.6	-	232.4±7.7	228.6±10.0
	Cost Rate	0.225	-	0.227	0.219
	Return	8.2±0.5	-	8.0±0.2	7.6±0.5

Table 8: Performance of the expert data.

Environment		Cost	Return
Safexp-PointGoal1-v0	Safe	8.0±8.29	19.5±2.8
	Unsafe	64.9±13.3	18.9±2.7
	Mixed	27.0±28.7	19.3±2.8
Safexp-PointButton1-v0	Safe	26.6±15.0	20.1±4.2
	Unsafe	164.8±49.3	19.0±3.0
	Mixed	72.7±72.1	19.8±4.0
Safexp-CarGoal1-v0	Safe	6.7±7.9	25.9±4.0
	Unsafe	82.6±36.90	22.6±3.2
	Mixed	32.0±42.1	24.8±4.1
Safexp-CarButton1-v0	Safe	139.2±41.7	23.8±5.4
	Unsafe	310.5±40.9	24.3±4.0
	Mixed	196.3±90.8	24.0±5.0
Safexp-DoggoGoal1-v0	Safe	24.9±12.4	21.0±3.7
	Unsafe	121.6±26.5	20.4±2.8
	Mixed	57.2±49.1	20.8±3.5
Safexp-DoggoButton1-v0	Safe	172.9±61.6	15.5±5.8
	Unsafe	349.3±35.0	14.4±3.6
	Mixed	231.7±99.3	15.1±5.2

Table 9: Performance of the expert that is evaluated with 100 trajectories.

Environment		Cost	Return
Safexp-PointGoal1-v0	Expert 1	27.8±18.5	15.6±3.2
	Expert 2	12.5±18.9	13.4±7.7
	Expert 3	26.1±27.9	17.9±4.6
Safexp-PointButton1-v0	Expert 1	54.8±40.2	13.4±5.3
	Expert 2	46.7±55.1	11.8±8.5
	Expert 3	70.6±56.7	13.9±6.4
Safexp-CarGoal1-v0	Expert 1	28.4±29.3	21.8±6.3
	Expert 2	20.7±25.9	21.6±7.6
	Expert 3	24.5±26.2	21.4±8.3
Safexp-CarButton1-v0	Expert 1	220.4±87.6	17.6±5.8
	Expert 2	220.0±117.9	19.8±7.1
	Expert 3	197.1±85.0	20.2±7.6
Safexp-DoggoGoal1-v0	Expert 1	54.2±45.2	15.0±5.8
	Expert 2	56.6±46.1	22.8±4.3
	Expert 3	57.2±42.2	18.1±5.1
Safexp-DoggoButton1-v0	Expert 1	218.0±95.1	9.5±5.2
	Expert 2	243.4±115.9	11.5±4.6
	Expert 3	256.8±107.4	13.0±6.4



(a) Safexp-DoggoGoal1-v0

(b) Safexp-DoggoButton1-v0

Figure 6: Learning curves in Safexp-DoggoGoal1-v0 and Safexp-DoggoButton1-v0. Performance is measured with Cost, Cost Rate, and Return. The x-axis represents time steps of interactions with the environment. Each algorithm is evaluated with 5 random seeds.

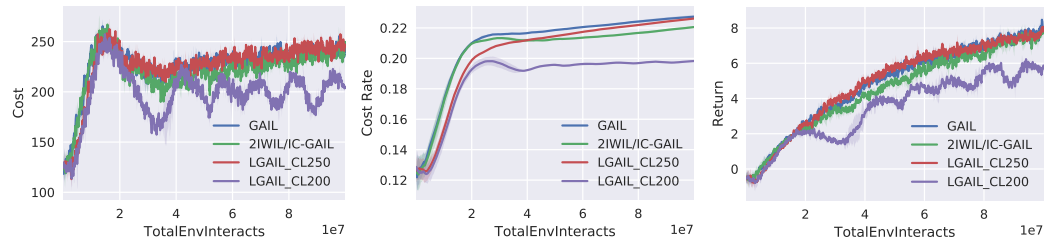


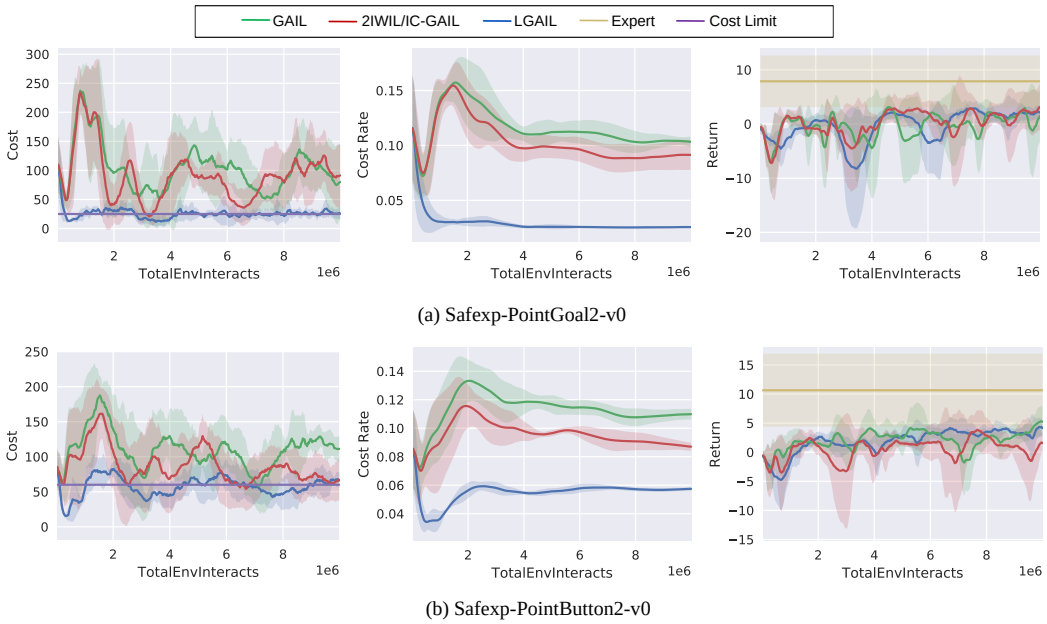
Figure 7: Learning curves in Safexp-DoggoButton1-v0 with different cost limits.

C.4 Experiments on "Level 2" Tasks

We conduct experiments on "level 2" tasks (Safexp-PointGoal2-v0 and Safexp-PointButton2-v0) to demonstrate the performance of LGAIL against other baselines. The learning curves are presented in Figure 8. In more complex environments, performance degradation is observed for experts and IL algorithms. However, experiment results show that LGAIL can work effectively in these complex environments, i.e., LGAIL is able to achieve the same level of performance regarding Return compared with GAIL and 2IWIL/IC-GAIL and simultaneously satisfy the cost limit.

C.5 Impact of Cost Limits

In the paper, we carry out experiments to investigate the impact of cost limits on LGAIL's performance with only unsafe expert data. We only present the results using Safexp-PointGoal1-v0 in the paper. Here, more experiments in other environments are given, which are shown in Figure 9. We can see that LGAIL is able to obtain a policy that satisfies the given cost limit.



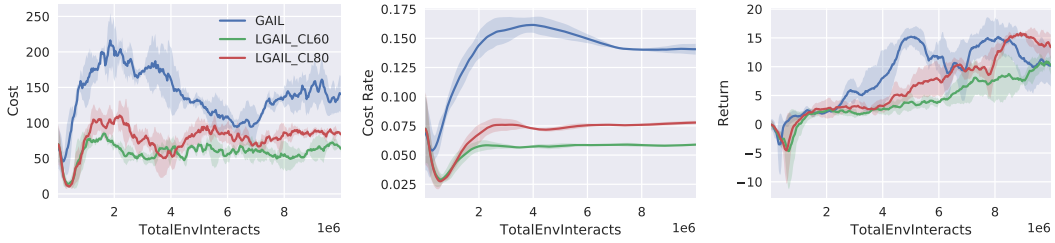
(a) Safexp-PointGoal2-v0

(b) Safexp-PointButton2-v0

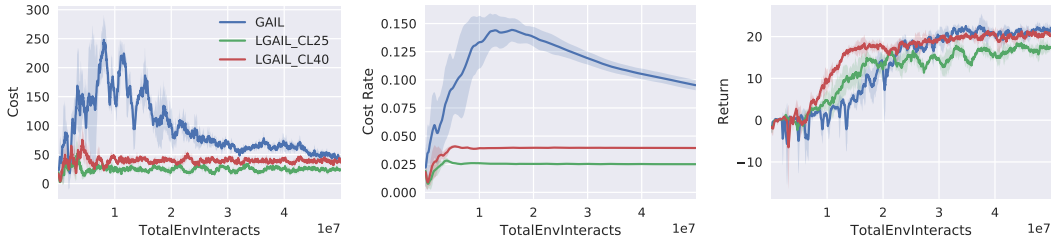
Figure 8: Learning curves of LGAIL and the other baselines on Level 2 tasks. Performance is measured with Cost, Cost Rate, and Return. The x-axis represents time steps of interactions with the environment. Each algorithm is evaluated with 5 random seeds.

C.6 GAIL WITH PURELY SAFE EXPERT DATA

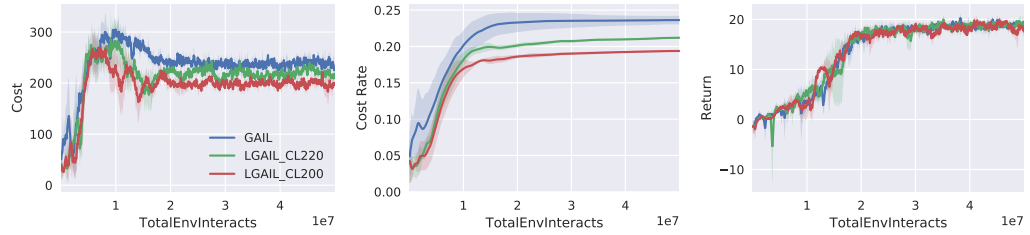
We conduct experiments to investigate the impact of the number of purely safe expert data as well as the diversity of expert data on the performance of GAIL. Concretely, in environments Safexp-PointGoal1-v0 and Safexp-PointButton1-v0, we train GAIL with different numbers of expert trajectories (including 10, 30, 100, 300, and 1000 trajectories). These expert data are sampled from a single expert. Each trajectory contains 1,000 state-action pairs. Hence, it means that one million safe state-action pairs are provided when we use 1000 trajectories to train GAIL, which is a huge amount of data. Besides, we also use purely safe expert data that are sampled from three independent experts to train an agent. For the experiments that use data sampled from a mixture of expert policies, we evaluate the performance of GAIL against different numbers of expert trajectories (300, 900, and 1500). The learning curves are presented in Figure 10. From the figures, we can see that GAIL usually fails to recover a safe policy even with abundant purely safe expert data. For example, in Safexp-PointGoal1-v0 with expert data from one expert, GAIL might manage to recover a safe policy, but it is not guaranteed and could be affected by the expert data. In Safexp-PointButton1-v0, GAIL fails to recover safe policies from purely safe expert data. Besides, we find that increasing the number of expert data does not help improve the performance of GAIL in terms of both rewards and costs. What's worse, GAIL performs worse with expert data that are sampled from multiple experts. In practice, expert data are often collected from various sources. As a result, GAIL is not enough to solve the Safe IL task, whereas our algorithm LGAIL takes safety into consideration and is guaranteed to generate safe policies.



(a) Safexp-PointButton1-v0

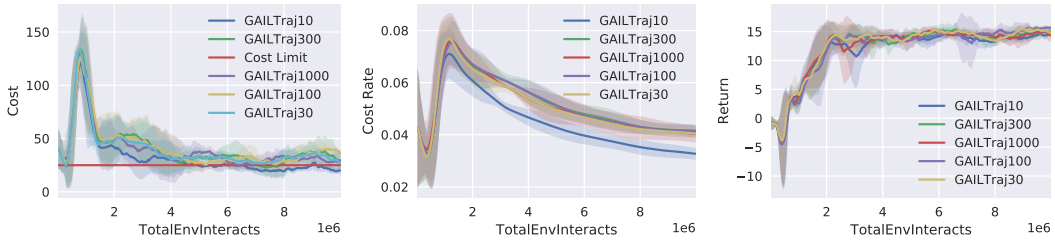


(b) Safexp-CarGoal1-v0

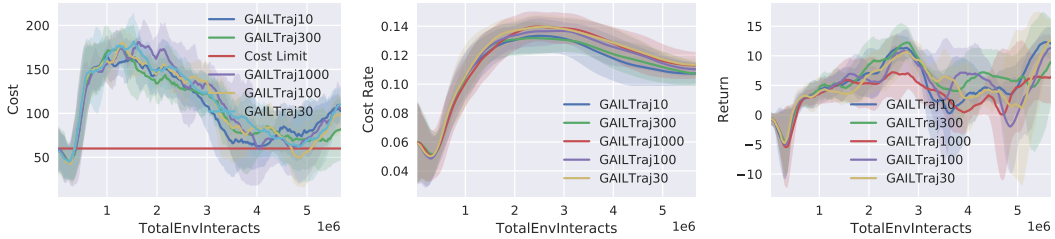


(c) Safexp-CarButton1-v0

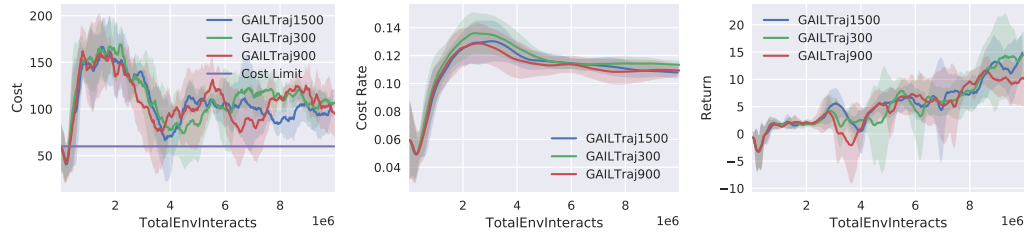
Figure 9: Impact of constraint limit on LGAIL with purely unsafe expert data. In the legend LGAIL_CL$\{x\}$, x represents the cost limit d_0.



(a) Safexp-PointGoal1-v0 (purely safe expert data sampled from one expert)



(b) Safexp-PointButton1-v0 (purely safe expert data sampled from one expert)



(c) Safexp-PointButton1-v0 (purely safe expert data sampled from three experts)

Figure 10: Impact of the number of expert data on GAIL with purely safe expert data. In the legend GAILTraj$\{x\}$, x represents the number of expert trajectories.