
Imitation Learning from Observations by Minimizing Inverse Dynamics Disagreement

Xiaojian Ma^{12*}, Chao Yang^{1*}, Wenbing Huang^{1*},

Fuchun Sun¹, Huaping Liu¹, Junzhou Huang³, Chuang Gan⁴

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

² Center for Vision, Cognition, Learning and Autonomy, Department of Computer Science, UCLA

³ Tencent AI Lab

⁴ MIT-IBM Watson AI Lab

maxiaojian@ucla.edu, yangchao18@mails.tsinghua.edu.cn

hwenbing@126.com, fcsun@tsinghua.edu.cn

Abstract

This paper studies *Learning from Observations (LfO)* for imitation learning with access to state-only demonstrations. In contrast to *Learning from Demonstration (LfD)* that involves both action and state supervision, LfO is more practical in leveraging previously inapplicable resources (*e.g.* videos), yet more challenging due to the incomplete expert guidance. In this paper, we investigate LfO and its difference with LfD in both theoretical and practical perspectives. We first prove that the gap between LfD and LfO actually lies in the disagreement of inverse dynamics models between the imitator and the expert, if following the modeling approach of GAIL [16]. More importantly, the upper bound of this gap is revealed by a negative causal entropy which can be minimized in a model-free way. We term our method as *Inverse-Dynamics-Disagreement-Minimization* (IDDM) which enhances the conventional LfO method through further bridging the gap to LfD. Considerable empirical results on challenging benchmarks indicate that our method attains consistent improvements over other LfO counterparts.

1 Introduction

A crucial aspect of intelligent robots is their ability to perform a task of interest by imitating expert behaviors from raw sensory observations [5]. Towards this goal, GAIL [16] is one of the most successful imitation learning methods, which adversarially minimizes the discrepancy of the occupancy measure between the agent and the expert for policy optimization. However, along with many other methods [33, 3, 31, 32, 1, 25, 6, 10, 14, 2], GAIL adopts a heavily supervised training mechanism, which demands not only the expert’s state (*e.g.* observable spatial locations), but also its accurate action (*e.g.* controllable motor commands) performed at each time step.

Whereas providing expert action indeed enriches the information and hence facilitates the imitation learning process, collecting them could be difficult and sometimes infeasible for some certain practical cases, particularly when we would like to learn skills from a large number of internet videos. Besides, imitation learning under action guidance is not biologically reasonable [42], as our human can imitate skills through adjusting the action to match the demonstrators’ state, without knowing what exact action the demonstrator has performed. To address these concerns, several methods have been

*Denotes equal contributions. Corresponding author: Fuchun Sun.

proposed [37, 42, 5, 23, 38], including the one named GAIfO [43] that extends the idea of GAIL to the case with the absence of action guidance. Joining the previous denotations, this paper will define the original problem as *Learning from Demonstrations (LfD)*, and the new action agnostic setting as *Learning from Observations (LfO)*.

Undoubtedly, conducting LfO is *non-trivial*. For many tasks (*e.g.* robotic manipulation, locomotion and video-game playing), the reward function depends on both action and state. It remains challenging to determine the optimal action corresponding to the best reward purely from experts' state observations, since there could be multiple choices of action corresponding to the same sequence of state in a demonstration—when, for example, manipulating redundant-degree robotic hands, there exist countless force controls of joints giving rise to the same pose change. Yet, realizing LfO is still possible, especially if the expert and the agent share the same dynamics system (namely, the same robot). In this condition and what this paper has assumed, the correlation between action and state can be learned by the self-playing of the agent (see for example in [42]).

In this paper, we approach LfO by leveraging the concept of *inverse dynamics disagreement minimization*. As its name implies, inverse dynamics disagreement is defined as the discrepancy between the inverse dynamics models of the expert and the agent. Minimizing such disagreement becomes the task of inverse dynamics prediction, a well-known problem that has been studied in robotics [26]. Interestingly, as we will draw in this paper, the inverse dynamics disagreement is closely related to LfD and LfO. To be more specific, we prove that the inverse dynamics disagreement actually accounts for the optimization gap between LfD and naive LfO, if we model LfD by using GAIL [16] and consider naive LfO as GAIfO [43]. This result is crucial, not only for it tells the quantitative difference between LfD and naive LfO but also for it enables us to solve LfO more elegantly by minimizing the inverse dynamics disagreement as well.

To mitigate the issue of inverse dynamics disagreement, here we propose a model-free solution for the consideration of efficiency. In detail, we derive an upper bound of the gap, which turns out to be a negative entropy of the state-action occupancy measure. Under the assumption of deterministic system, such entropy contains a mutual information term that can be optimized with the popularly-used tool (*i.e.* MINE [4]). For convenience, we term our method as the *Inverse-Dynamics-Disagreement-Minimization* (IDDM) based LfO in what follows. To verify the effectiveness of our IDDM, we perform experimental comparisons on seven challenging control tasks, ranging from traditional control to locomotion [9]. The experimental results demonstrate that our proposed method attains consistent improvements over other LfO counterparts.

The rest of the paper is organized as follows. In Sec. 2, we will first review some necessary notations and preliminaries. Then our proposed method will be detailed in Sec. 3 with theoretical analysis and efficient implementation, and the discussions with existing LfD and LfO methods will be included in Sec. 4. Finally, experimental evaluations and ablation studies will be demonstrated in Sec. 5.

2 Preliminaries

Notations. To model the action decision procedure in our context, we consider a standard Markov decision process (MDP) [39] as $(\mathcal{S}, \mathcal{A}, r, \mathcal{T}, \mu, \gamma)$, where \mathcal{S} and \mathcal{A} are the sets of feasible state and action, respectively; $r(s, a) : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ denotes the reward function on state s and action a ; $\mathcal{T}(s'|s, a) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ characterizes the dynamics of the environment and defines the transition probability to next-step state s' if the agent takes action a at current state s ; $\mu(s) : \mathcal{S} \rightarrow [0, 1]$ is the distribution of initial state and $\gamma \in (0, 1)$ is the discount factor. A stationary policy $\pi(a|s) : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ defines the probability of choosing action a at state s . A temporal sequence of state-action pairs $\{(s_0, a_0), (s_1, a_1), \dots\}$ is called a trajectory denoted by ζ .

Occupancy measure. To characterize the statistical properties of an MDP, the concept of occupancy measure [30, 40, 16, 18] is proposed to describe the distribution of state and action under a given policy π . Below, we introduce its simplest form, *i.e.*, *State Occupancy Measure*.

Definition 1 (State Occupancy Measure). *Given a stationary policy π , state occupancy measure $\rho_\pi(s) : \mathcal{S} \rightarrow \mathbb{R}$ denotes the discounted state appearance frequency under policy π*

$$\rho_\pi(s) = \sum_{t=0}^{\infty} \gamma^t P(s_t = s | \pi). \quad (1)$$

With the use of state occupancy measure, we can define other kinds of occupancy measures under different supports, including state-action occupancy measure, station transition occupancy measure, and joint occupancy measure. We list their definitions in Tab. 1 for reader’s reference.

Inverse dynamics model. We present the *inverse dynamics model* [36, 35] in Definition 2, which infers the action inversely given state transition (s, s') .

Table 1: Different occupancy measures for MDP

	State-Action Occupancy Measure	State Transition Occupancy Measure	Joint Occupancy Measure
Denotation	$\rho_\pi(s, a)$	$\rho_\pi(s, s')$	$\rho_\pi(s, a, s')$
Support	$\mathcal{S} \times \mathcal{A}$	$\mathcal{S} \times \mathcal{S}$	$\mathcal{S} \times \mathcal{A} \times \mathcal{S}$
Definition	$\rho_\pi(s)\pi(a s)$	$\int_{\mathcal{A}} \rho_\pi(s, \bar{a})\mathcal{T}(s' s, \bar{a})d\bar{a}$	$\rho_\pi(s, a)\mathcal{T}(s' s, a)$

Definition 2 (Inverse Dynamics Model). Let $\rho_\pi(a|s, s')$ denotes the density function of the inverse dynamics model under the policy π , whose relation with \mathcal{T} and π can be shown as follows.

$$\rho_\pi(a|s, s') := \frac{\mathcal{T}(s'|s, a)\pi(a|s)}{\int_{\mathcal{A}} \mathcal{T}(s'|s, \bar{a})\pi(\bar{a}|s)d\bar{a}}. \quad (2)$$

3 Methodology

In this section, we first introduce the concepts of LfD, naive LfO, and *inverse dynamics disagreement*. Then, we prove that the optimization gap between LfD and naive LfO actually leads to the inverse dynamics disagreement. As such, we enhance naive LfO by further minimizing the inverse dynamics disagreement. We also demonstrate that such disagreement can be bounded by an entropy term and can be minimized by a model-free method. Finally, we provide a practical implementation for our proposed method.

3.1 Inverse Dynamics Disagreement: the Gap between LfD and LfO

LfD. In Sec. 1, we have mentioned that GAIL and many other LfD methods [16, 19, 17, 29] exploit the discrepancy of the occupancy measure between the agent and expert as a reward for policy optimization. Without loss of generality, we will consider GAIL as the representative LfD framework and build our analysis on this description. This LfD framework requires to compute the discrepancy over the state-action occupancy measure, leading to

$$\min_{\pi} \mathbb{D}_{\text{KL}}(\rho_\pi(s, a) || \rho_E(s, a)), \quad (3)$$

where $\rho_E(s, a)$ denotes the occupancy measure under the expert policy, and $\mathbb{D}_{\text{KL}}(\cdot)$ computes the Kullback-Leibler (KL) divergence². We have omitted the policy entropy term in GAIL, but our following derivations will find that the policy entropy term is naturally contained in the gap between LfD and LfO.

Naive LfO. In LfO, the expert action is absent, thus directly working on $\mathbb{D}_{\text{KL}}(\rho_\pi(s, a) || \rho_E(s, a))$ is infeasible. An alternative objective could be minimizing the discrepancy on the state transition occupancy measure $\rho_\pi(s, s')$, as mentioned in GAIIfO [43]. The objective function in (3) becomes

$$\min_{\pi} \mathbb{D}_{\text{KL}}(\rho_\pi(s, s') || \rho_E(s, s')). \quad (4)$$

²The original GAIL method applies Jensen-Shannon (JS) divergence rather than KL divergence for measurement. Here, we will use KL divergence for the consistency throughout our derivations. Indeed, our method is also compatible with JS divergence, with the details provided in the supplementary material.

We will refer this as *naive LfO* in the following context. Compared to LfD, the key challenge in LfO comes from the absence of action information, which prevents it from applying typical action-involved imitation learning approaches like behavior cloning [33, 3, 12, 31, 32] or apprenticeship learning [25, 1, 40]. Actually, action information can be implicitly encoded in the state transition (s, s') . We have assumed the expert and the agent share the same dynamics system $\mathcal{T}(s'|s, a)$. It is thus possible for us to learn the action-state relation by exploring the difference between their inverse dynamics models.

We define the inverse dynamics disagreement between the expert and the agent as follows.

Definition 3 (Inverse Dynamics Disagreement). *Given expert policy π_E and agent policy π , the inverse dynamics disagreement is defined as the KL divergence between the inverse dynamics models of the expert and the agent.*

$$\text{Inverse Dynamics Disagreement} := \mathbb{D}_{\text{KL}}(\rho_{\pi}(a|s, s') || \rho_{E}(a|s, s')). \quad (5)$$

Given a state transition (s, s') , minimizing the inverse dynamics disagreement is learning an optimal policy to fit the expert/ground-truth action labels. This is a typical robotic task [26], and it can be solved by using a mixture method of combining machine learning model and control model.

Here, we contend another role of the inverse dynamics disagreement in the context of imitation learning. Joining the denotations in (3), (4) and Definition 3, we provide the following result.

Theorem 1. *If the agent and the expert share the same dynamics system, the relation between LfD, naive LfO, and inverse dynamics disagreement can be characterized as*

$$\mathbb{D}_{\text{KL}}(\rho_{\pi}(a|s, s') || \rho_{E}(a|s, s')) = \mathbb{D}_{\text{KL}}(\rho_{\pi}(s, a) || \rho_{E}(s, a)) - \mathbb{D}_{\text{KL}}(\rho_{\pi}(s, s') || \rho_{E}(s, s')). \quad (6)$$

Theorem 1 states that the inverse dynamics disagreement essentially captures the optimization gap between LfD and naive LfO. As (5) is non-negative by nature, optimizing the objective of LfD implies minimizing the objective of LfO but not vice versa. One interesting observation is that when the action corresponding to a given state transition is unique (or equivalently, the dynamics $\mathcal{T}(s'|s, a)$ is injective w.r.t a), the inverse dynamics is invariant to different conducted policies, hence the inverse dynamics disagreement between the expert and the agent reduces to zero. We summarize this by the following corollary.

Corollary 1. *If the dynamics $\mathcal{T}(s'|s, a)$ is injective w.r.t a , LfD is equivalent to naive LfO.*

$$\mathbb{D}_{\text{KL}}(\rho_{\pi}(s, a) || \rho_{E}(s, a)) = \mathbb{D}_{\text{KL}}(\rho_{\pi}(s, s') || \rho_{E}(s, s')). \quad (7)$$

However, since most of the real world tasks are performed in rather complex environments, (5) is usually not equal to zero and the gap between LfD and LfO should not be overlooked, which makes minimizing the inverse dynamics disagreement become unavoidable.

3.2 Bridging the Gap with Entropy Maximization

We have shown that the inverse dynamics disagreement amounts to the optimization gap between LfD and naive LfO. Therefore, the key to improving naive LfO mainly lies in *inverse dynamics disagreement minimization*. Nevertheless, accurately computing the disagreement is difficult, as it relies on the environment dynamics \mathcal{T} and the expert policy (see (2)), both of which are assumed to be unknown. In this section, we try a smarter way and propose an upper bound for the gap, without the access of the dynamics model and expert guidance. This upper bound is tractable to be minimized if assuming the dynamics to be deterministic. We introduce the upper bound by the following theorem.

Theorem 2. *Let $\mathcal{H}_{\pi}(s, a)$ and $\mathcal{H}_E(s, a)$ denote the causal entropies over the state-action occupancy measure of the agent and expert, respectively. When $\mathbb{D}_{\text{KL}}[\rho_{\pi}(s, s') || \rho_{E}(s, s')]$ is minimized, we have*

$$\mathbb{D}_{\text{KL}}[\rho_{\pi}(a|s, s') || \rho_{E}(a|s, s')] \leq -\mathcal{H}_{\pi}(s, a) + \text{Const}. \quad (8)$$

Now we take a closer look at $\mathcal{H}_{\pi}(s, a)$. Following the definition in Tab. 1, the entropy of state-action occupancy measure can be decomposed as the sum of the policy entropy and the state entropy by

$$\begin{aligned} \mathcal{H}_{\pi}(s, a) &= \mathbb{E}_{\rho_{\pi}(s, a)}[-\log \rho_{\pi}(s, a)] = \mathbb{E}_{\rho_{\pi}(s, a)}[-\log \pi(a|s)] + \mathbb{E}_{\rho_{\pi}(s)}[-\log \rho_{\pi}(s)] \\ &= \mathcal{H}_{\pi}(a|s) + \mathcal{H}_{\pi}(s). \end{aligned} \quad (9)$$

For the first term, the policy entropy $\mathcal{H}_{\pi}(a|s)$ can be estimated via sampling similar to previous studies [16]. For the second term, we leverage the mutual information (MI) between s and (s', a) to obtain an unbiased estimator of the entropy over the state occupancy measure, namely,

Algorithm 1 Inverse-Dynamics-Disagreement-Minimization (IDDM)

Input: State-only expert demonstrations $\mathcal{D}_E = \{\zeta_i^E\}$ where $\zeta_i = \{s_0^E, s_1^E, \dots\}$, policy π_θ , discriminator D_ϕ , MI estimator \mathcal{I} , entropy weights λ_p, λ_s , maximum iterations M .

for 1 to M **do**

 Sample agent rollouts $\mathcal{D}_A = \{\zeta^i\}$, $\zeta^i \sim \pi_\theta$ and update the MI estimator \mathcal{I} with \mathcal{D}_A .
 Update the discriminator D_ϕ with the gradient

$$\hat{\mathbb{E}}_{\mathcal{D}_A} [\nabla_\phi \log D_\phi(s, s')] + \hat{\mathbb{E}}_{\mathcal{D}_E} [\nabla_\phi \log(1 - D_\phi(s, s'))].$$

 Update policy π_θ using the following gradient (can be integrated into methods like PPO [34])

$$\hat{\mathbb{E}}_{\mathcal{D}_A} [\nabla_\theta \log \pi_\theta(a|s) Q(s, a)] - \lambda_p \nabla_\theta \mathcal{H}_{\pi_\theta}(a|s) - \lambda_s \nabla_\theta \mathcal{I}_{\pi_\theta}(s; (s', a)),$$

$$\text{where } Q(\bar{s}, \bar{a}) = \hat{\mathbb{E}}_{\mathcal{D}_A} [\log D_\phi(s, s') | s_0 = \bar{s}, a_0 = \bar{a}].$$

end for

$$\mathcal{H}_\pi(s) = \mathcal{I}_\pi(s; (s', a)) + \underbrace{\mathcal{H}_\pi(s|s', a)}_{=0} = \mathcal{I}_\pi(s; (s', a)), \quad (10)$$

where we have $\mathcal{H}_\pi(s|s', a) = 0$ as we have assumed $(s, a) \rightarrow s'$ is a deterministic function³. In our implementation, the MI $\mathcal{I}_\pi(s; (s', a))$ is computed via maximizing the lower bound of KL divergence between the product of marginals and the joint distribution following the formulation of [27]. Specifically, we adopt MINE [4, 15] which implements the score function with a neural network to achieve a low-variance MI estimator.

Overall loss. By combining the results in Theorem 1, Theorem 2, (9) and (10), we enhance naive LfO by further minimizing the upper bound of its gap to LfD. The eventual objective is

$$\mathcal{L}_\pi = \mathbb{D}_{\text{KL}}(\rho_\pi(s, s') || \rho_E(s, s')) - \lambda_p \mathcal{H}_\pi(a|s) - \lambda_s \mathcal{I}_\pi(s; (s', a)), \quad (11)$$

where the first term is from naive LfO, and the last two terms are to minimize the gap between LfD and naive LfO. We also add trade-off weights λ_p and λ_s to the last two terms for more flexibility.

Implementation. As our above derivations can be generalized to JS-divergence (see Sec. A.3-4 in the supplementary material), we can utilize the GAN-like [27] method to minimize the first term in (11). In detail, we introduce a parameterized discriminator network D_ϕ and a policy network π_θ (serves as a generator) to realize the first term in (11). The term $\log D_\phi(s, s')$ could be interpreted as an immediate cost since we minimize its expectation over the current occupancy measure. A similar training method can also be found in GAIL [16], but it relies on state-action input instead. We defer the derivations for the gradients of the causal entropy $\nabla \mathcal{H}_\pi(a|s)$ and MI $\nabla \mathcal{I}_\pi(s; (s', a))$ with respect to the policy in Sec. A.5 of the supplementary material. Note that the objective (11) can be optimized by any policy gradient method, like A3C [24] or PPO [34], and we apply PPO in our experiments. The algorithm details are summarized in Alg. 1.

4 Related Work

4.1 Learning from Demonstrations

Modern dominant approaches on LfD mainly fall into two categories: **Behavior Cloning (BC)** [33, 3, 31, 32], which seeks the best policy that can minimize the action prediction error in demonstration, and **Inverse Reinforcement Learning (IRL)** [25, 1], which infers the reward used by expert to guide the agent policy learning procedure. A notable implementation of the latter is GAIL [16], which reformulates IRL as an occupancy measure matching problem [30], and utilizes the GAN [13] method

³In this paper, the tasks in our experiments indeed reveal deterministic dynamics. The mapping $(s, a) \rightarrow s'$ is deterministic also underlying that $(s', a) \rightarrow s$ is deterministic. Referring to [22], when s, s', a are continuous, $\mathcal{H}(s|s', a)$ can be negative; but since these variables are actually quantified as finite-bit precision numbers (e.g. stored as 32-bit discrete numbers in computer), it is still true that conditional entropy is zero in practice.

along with a forward RL to minimize the discrepancy of occupancy measures between imitator and demonstrator. There are also several follow-up works that attempt to enhance the effectiveness of discrepancy computation [20, 17, 11, 29], whereas all these methods require exact action guidance at each time step.

4.2 Learning from Observations

There have already been some researches on exploring LfO. These approaches exploit either a complex hand-crafted reward function or an inverse dynamics model that predicts the exact action given state transitions. Here is a summary to show how they are connected to our method.

LfO with Hand-crafted Reward and Forward RL. Recently, Peng et al. propose **DeepMimic**, a method that can imitate locomotion behaviors from motion clips without action labeling. They design a reward to encourage the agent to directly match the expert’s physical proprieties, such as joint angles and velocities, and run a forward RL to learn the imitation policy. However, as the hand-crafted reward function does not take expert action (or implicitly state transition) into account, it is hard to be generalized to tasks whose reward depends on actions.

Model-Based LfO. **BCO** [42] is another LfO approach. The authors infer the exact action from state transition with a learned inverse dynamics model (2). The state demonstrations augmented with the predicted actions deliver common state-action pairs that enable imitation learning via BC [33]. At its heart, the inverse dynamics model is trained in parallel by collecting rollouts in the environment. However, as showed in (2), the inverse dynamics model depends on the current policy, underlying that an optimal inverse dynamics model would be infeasible to obtain before the optimal policy is learned. The performance of BCO would thus be not theoretically guaranteed.

LfO with GAIL. **GAIfO** [43, 44] is the closest work to our method. The authors follow the formulation of GAIL [16] but replace the state-action definition (s, a) with state transition (s, s') , which gives the same objective in Eq. (4) if replacing KL with JS divergence. As we have discussed in Sec. 3.1, there is a gap between Eq. (4) and the objective of original LfD in Eq. (3), and this gap is induced by inverse dynamics disagreement. Unlike our method, the solution by GAIfO never minimizes the gap and is thereby no better than ours in principal.

5 Experiments

For the experiments below, we investigate the following questions:

1. Does inverse dynamics disagreement really account for the gap between LfD and LfO?
2. With state-only guidance, can our method achieve better performance than other counterparts that do not consider inverse dynamics disagreement minimization?
3. What are the key ingredients of our method that contribute to performance improvement?

To answer the first question, we conduct toy experiments with the *Gridworld* environment [39]. We test and contrast the performance of our method (refer to Eq. (11)) against GAIL (refer to Eq. (3)) and GAIfO (refer to Eq. (4)) on the tasks under different levels of inverse dynamics disagreement. Regarding the second question, we evaluate our method against several baselines on six physics-based control benchmarks [9], ranging from low-dimension control to challenging high-dimension continuous control. Finally, we explore the ablation analysis of two major components in our method (the policy entropy term and the MI term) to address the last question. Due to the space limit, we defer more detailed specifications of all the evaluated tasks into the supplementary material.

5.1 Understanding the Effect of Inverse Dynamics Disagreement

This collection of experiments is mainly to demonstrate how inverse dynamics disagreement influences the LfO approaches. We first observe that inverse dynamics disagreement will increase when the number of possible action choices grows. This is justified in Fig. 1a, and more details about the relation between inverse dynamics disagreement and the number of action choices are provided in Sec. B.2 of the supplementary material. Hence, we can utilize different action scales to reflect

different levels of inverse dynamics disagreement in our experiments. Controlling the action scale in *Gridworld* is straightforward. For example in Fig. 1b, agent block (in red) may take various kinds of actions (walk, jump or others) for moving to a neighbor position towards the target (in green), and we can specify different numbers of action choices.

We simulate the expert demonstrations by collecting the trajectories of the policy trained by PPO [34]. Then we conduct GAIL, GAIfO, and our method, and evaluate the pre-defined reward values for the policies they learn. It should be noted that all imitation learning methods have no access to the reward function during training. As we can see in Fig. 1c, the gap between GAIL and GAIfO is growing as the number of action choices (equivalently the level of inverse dynamics disagreement) increases, which is consistent with our conclusion in Theorem 1. We also find that the rewards of GAIL and GAIfO are the same when the number of action choice is 1 (*i.e.* the dynamics is injective), which follows the statement in Corollary 1. Our method lies between GAIL and GAIfO, indicating that the gap between GAIL and GAIfO can be somehow mitigated by explicitly minimizing inverse dynamics disagreement. Note that, GAIL also encounters performance drop when inverse dynamics disagreement becomes large. This is mainly because the imitation learning problem itself also becomes more difficult when the dynamics is complicated and beyond injective.

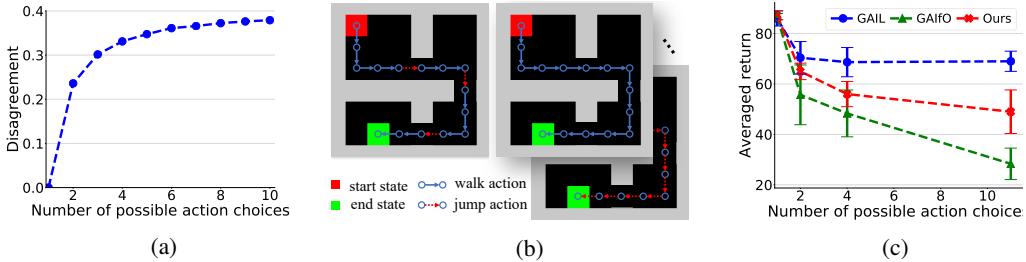


Figure 1: Toy examples on illustrating the effect of inverse dynamics disagreement.

5.2 Comparative Evaluations

For comparative evaluations, we carry out several LfO baselines, including DeepMimic [28], BCO [42], and GAIfO [43]. In particular, we introduce a modified version of GAIfO that only takes a single state as input to illustrate the necessity of leveraging state transition; we denote this method as GAIfO-s. We also run GAIL [16] to provide oracle reference. All experiments are evaluated within fixed steps. On each task, we run each algorithm over five times with different random seeds. In Fig. 2, the solid curves correspond to the mean returns, and the shaded regions represent the variance over the five runs. The eventual results are summarized in Tab. 2, which is averaged over 50 trials of the learned policies. Due to the space limit, we defer more details to the supplementary material.

Table 2: Summary of quantitative results. All results correspond to the original exact reward defined in [7]. *CartPole* is excluded from DeepMimic because no crafting reward is available.

	CartPole	Pendulum	DoublePendulum	Hopper	HalfCheetah	Ant
DeepMimic	-	731.0±19.0	454.4±154.0	2292.6±1068.9	202.6±4.4	-985.3±13.6
BCO	200.0±0.0	24.9±0.8	80.3±13.1	1266.2±1062.8	4557.2±90.0	562.5±384.1
GAIfO	197.5±7.3	980.2±3.0	4240.6±4525.6	1021.4±0.6	3955.1±22.1	-1415.0±161.1
GAIfO-s*	200.0±0.0	952.1±23.0	1089.2±51.4	1022.5±0.40	2896.5±53.8	-5062.3±56.9
Ours	200.0±0.0	1000.0±0.0	9359.7±0.2	3300.9±52.1	5699.3±51.8	2800.4±14.0
GAIL	200.0±0.0	1000.0±0.0	9174.8±1292.5	3249.9±34.0	6279.0±56.5	5508.8±791.5
Expert	200.0±0.0	1000.0±0.0	9318.8±8.5	3645.7±181.8	5988.7±61.8	5746.8±117.5

*GAIfO with single state only.

The results read that our method achieves comparable performances with the baselines on the easy tasks (such as *CartPole*) and outperforms them by a large margin on the difficult tasks (such as *Ant*, *Hopper*). We also find that our algorithm exhibits more stable behaviors. For example, the

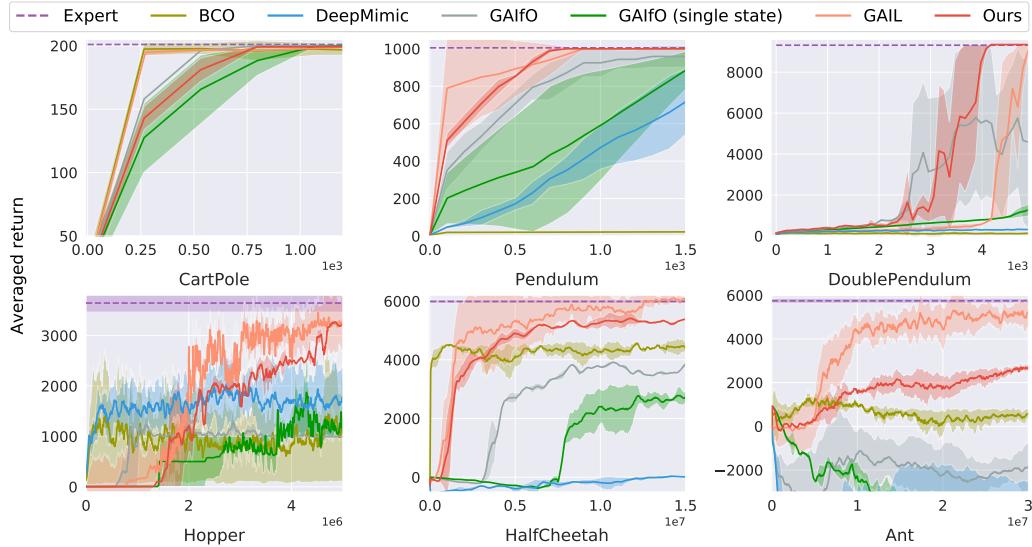


Figure 2: Learning curves under challenging robotic control benchmarks. For each experiment, a step represents one interaction with the environment. Detailed plots can be found in the supplementary.

performance of BCO on *Ant* and *Hopper* will unexpectedly drop down as the training continues. We conjecture that BCO explicitly but not accurately learns the inverse dynamics model from data, which yet is prone to over-fitting and leads to performance degradation. Conversely, our algorithm is model-free and guarantees the training stability as well as the eventual performance, even for the complex tasks including *HalfCheetah*, *Ant* and *Hopper*.

Besides, GAIIfO performs better than GAIIfO-s in most of the evaluated tasks. This illustrates the importance of taking state-transition into account to reflect action information in LfO. Compared with GAIIfO, our method clearly attains consistent and significant improvements on *HalfCheetah* (+1744.2), *Ant* (+4215.0) and *Hopper* (+2279.5), thus convincingly verifying that minimizing the optimization gap induced by inverse dynamics disagreement plays an essential role in LfO, and our proposed approach can effectively bridge the gap. For the tasks that have relatively simple dynamics (*e.g.* *CartPole*), GAIIfO achieves satisfying performances, which is consistent with our conclusion in Corollary 1.

DeepMimic that relies on hand-crafted reward struggles on most of the evaluated tasks. Our proposed method does not depend on any manually-designed reward signal, thus it becomes more self-contained and more practical in general applications.

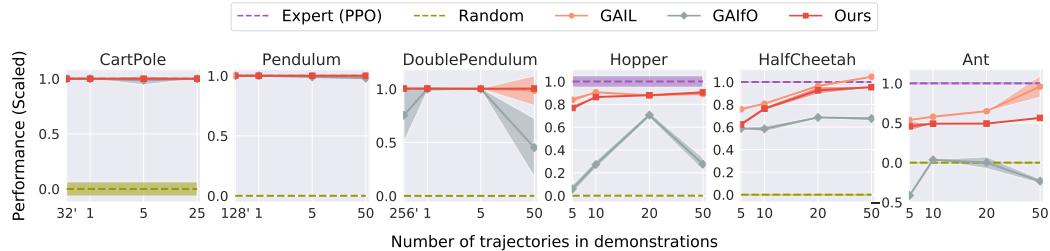


Figure 3: Comparative results of GAIL [16], GAIIfO [43] and our method with different number of trajectories in demonstrations. The performance is the averaged cumulative return over 5 trajectories and has been scaled within [0, 1] (the random and the expert policies are fixed to be 0 and 1, respectively). We also conduct experiments with demonstrations containing state-action/state-transition pairs with the number less than that within one complete trajectory. We use 32', 128' and 256' pairs (denoted in the beginning of the x axes) for the first three tasks, respectively.

Finally, we compare the performances of GAIL, GAIIfO and our method with different numbers of demonstrations. The results are presented in Fig. 3. It reads that for simple tasks like *CartPole* and *Pendulum*, there are no significant differences for all evaluated approaches, when the number of

demonstrations changes. While for the tasks with a higher dimension of state and action, our method performs advantageously over GAIfO. Even compared with GAIL that involves action demonstrations, our method still delivers comparable results. For all methods, more demonstrations facilitate better performances especially when the tasks become more complicated (*HalfCheetah* and *Ant*).

5.3 Ablation Study

The results presented in the previous section suggest that our proposed method can outperform other LfO approaches on several challenging tasks. Now we further perform a diverse set of analyses on assessing the impact of the policy entropy term and the MI term in (11). As these two terms are controlled by λ_p , λ_s , we will explore the sensitivity of our algorithm in terms of their values.

Sensitivity to Policy Entropy. We design four groups of parameters on *HalfCheetah*, where λ_p is selected from $\{0, 0.0005, 0.001, 0.01\}$ and λ_s is fixed at 0.01. The final results are plotted in Fig. 4, with the learning curves and detailed quantitative results provided in the supplementary material. The results suggest that we can always promote the performances by adding policy entropy. Although different choices of λ_p induce minor differences in their final performances, they are overall better than GAIfO that does not include the policy entropy term in its objective function.

Sensitivity to Mutual Information. We conduct four groups of experiments on *HalfCheetah* by ranging λ_s from 0.0 to 0.1 and fixing λ_p to be 0.001. The final results are shown in Fig. 5 (the learning curves and averaged return are also reported in the supplementary material). It is observed that the imitation performances could always benefit from adding the MI term, and the improvements become more significant when the λ_s has a relatively large magnitude. All of the variants of our method consistently outperform GAIfO, thus indicating the importance of the mutual information term in our optimization objective.

We also provide the results of performing a grid search on λ_s and λ_p in the supplementary material to further illustrate how better performance could be potentially obtained.

6 Conclusion

In this paper, our goal is to perform imitation Learning from Observations (LfO). Based on the theoretical analysis for the difference between LfO and Learning from Demonstrations (LfD), we introduce inverse dynamics disagreement and demonstrate it amounts to the gap between LfD and LfO. To minimize inverse dynamics disagreement in a principled and efficient way, we realize its upper bound as a particular negative causal entropy and optimize it via a model-free method. Our model, dubbed as *Inverse-Dynamics-Disagreement-Minimization* (IDDM), attains consistent improvement over other *LfO* counterparts on various challenging benchmarks. While our paper mainly focuses on control planning, further exploration on combining our work with representation learning to enable imitation across different domains could be a new direction for future work.

Acknowledgments

This research was jointly funded by Key (key grant) Project of Chinese Ministry of Education (Grant No.2018AAA0102900) and 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 Mingxuan Jing and Dr. Boqing Gong for the insightful discussions and the anonymous reviewers for the constructive feedback.

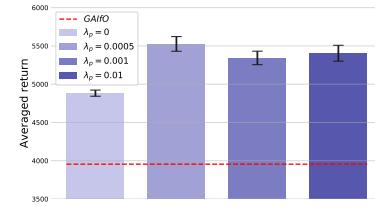


Figure 4: Sensitivity to the policy entropy weight λ_p .

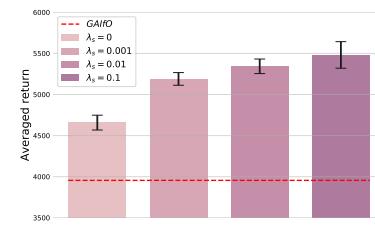


Figure 5: Sensitivity to the MI weight λ_s .

References

- [1] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *International conference on Machine learning (ICML)*, 2004.
- [2] Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *arXiv preprint arXiv:1808.00177*, 2018.
- [3] Christopher G Atkeson and Stefan Schaal. Robot learning from demonstration. In *International Conference on Machine Learning (ICML)*, 1997.
- [4] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International Conference on Machine Learning (ICML)*, 2018.
- [5] Darrin C Bentivegna, Christopher G Atkeson, and Gordon Cheng. Learning tasks from observation and practice. *Robotics and Autonomous Systems*, 47(2-3):163–169, 2004.
- [6] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.
- [7] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.
- [8] Erwin Coumans and Yunfei Bai. Pybullet, a python module for physics simulation in robotics, games and machine learning. 2016-2017.
- [9] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning (ICML)*, 2016.
- [10] Yan Duan, Marcin Andrychowicz, Bradly Stadie, Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. In *Advances in neural information processing systems (NeurIPS)*, 2017.
- [11] Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *International conference on Learning Representation (ICLR)*, 2018.
- [12] Alessandro Giusti, Jérôme Guzzi, Dan C Ciresan, Fang-Lin He, Juan P Rodríguez, Flavio Fontana, Matthias Faessler, Christian Forster, Jürgen Schmidhuber, Gianni Di Caro, et al. A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters (RA-L)*, 2016.
- [13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems (NeurIPS)*, 2014.
- [14] Tuomas Haarnoja, Aurick Zhou, Sehoon Ha, Jie Tan, George Tucker, and Sergey Levine. Learning to walk via deep reinforcement learning. In *Robotics: Science and Systems (RSS)*, 2019.
- [15] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations (ICLR)*, 2019.
- [16] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.
- [17] Mingxuan Jing, Xiaojian Ma, Wenbing Huang, Fuchun Sun, and Huaping Liu. Task transfer by preference-based cost learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2019.
- [18] Bingyi Kang, Zequn Jie, and Jiashi Feng. Policy optimization with demonstrations. In *International Conference on Machine Learning (ICML)*, 2018.
- [19] Beomjoon Kim and Joelle Pineau. Maximum mean discrepancy imitation learning. In *Robotics: Science and systems (RSS)*, 2013.
- [20] Kee-Eung Kim and Hyun Soo Park. Imitation learning via kernel mean embedding. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

- [21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International conference on Learning Representation (ICLR)*, 2015.
- [22] Rithesh Kumar, Anirudh Goyal, Aaron Courville, and Yoshua Bengio. Maximum entropy generators for energy-based models. *arXiv preprint arXiv:1901.08508*, 2019.
- [23] YuXuan Liu, Abhishek Gupta, Pieter Abbeel, and Sergey Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.
- [24] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning (ICML)*, 2016.
- [25] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *International conference on Machine learning (ICML)*, 2000.
- [26] Duy Nguyen-Tuong and Jan Peters. Model learning for robot control: a survey. *Cognitive processing*, 12(4):319–340, 2011.
- [27] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.
- [28] 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)*, 2018.
- [29] Xue Bin Peng, Angjoo Kanazawa, Sam Toyer, Pieter Abbeel, and Sergey Levine. Variational discriminator bottleneck: Improving imitation learning, inverse RL, and GANs by constraining information flow. In *International Conference on Learning Representations (ICLR)*, 2019.
- [30] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 1994.
- [31] Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2010.
- [32] Stephane Ross and J Andrew Bagnell. Agnostic system identification for model-based reinforcement learning. In *International conference on Machine learning (ICML)*, 2012.
- [33] Stefan Schaal. Learning from demonstration. In *Advances in neural information processing systems (NeurIPS)*, 1997.
- [34] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [35] Bruno Siciliano and Oussama Khatib. *Springer handbook of robotics*. Springer, 2008.
- [36] Mark W Spong and Romeo Ortega. On adaptive inverse dynamics control of rigid robots. *IEEE Transactions on Automatic Control (T-AC)*, 1990.
- [37] Bradly C Stadie, Pieter Abbeel, and Ilya Sutskever. Third-person imitation learning. In *International Conference on Learning Representations (ICLR)*, 2017.
- [38] Wen Sun, Anirudh Vemula, Byron Boots, and Drew Bagnell. Provably efficient imitation learning from observation alone. In *International Conference on Machine Learning (ICML)*, 2019.
- [39] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 1998.
- [40] Umar Syed, Michael Bowling, and Robert E Schapire. Apprenticeship learning using linear programming. In *International Conference on Machine Learning (ICML)*, 2008.
- [41] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012.
- [42] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.
- [43] Faraz Torabi, Garrett Warnell, and Peter Stone. Generative adversarial imitation from observation. *arXiv preprint arXiv:1807.06158*, 2018.
- [44] Faraz Torabi, Garrett Warnell, and Peter Stone. Imitation learning from video by leveraging proprioception. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.

A Proofs

A.1 Theorem 1 and its Corollary

Assumption 1. *The environment dynamics is stationary between the expert and agent.*

Lemma 1. *The equality below holds.*

$$\mathbb{D}_{\text{KL}} [\rho_\pi(s, a, s') || \rho_E(s, a, s')] = \mathbb{D}_{\text{KL}} [\rho_\pi(s, a) || \rho_E(s, a)]. \quad (12)$$

Proof. We can expand the left side of the equality following Kullback–Leibler divergence definition and Assumption 1 as

$$\begin{aligned} & \mathbb{D}_{\text{KL}} [\rho_\pi(s, a, s') || \rho_E(s, a, s')] \\ &= \mathbb{E}_{\rho_\pi} \left[\log \frac{\rho_\pi(s, a, s')}{\rho_E(s, a, s')} \right] \\ &= \mathbb{E}_{\rho_\pi} \left[\log \frac{\rho_\pi(s, a) \mathcal{T}(s'|s, a)}{\rho_E(s, a) \mathcal{T}(s'|s, a)} \right] \\ &= \mathbb{D}_{\text{KL}} [\rho_\pi(s, a) || \rho_E(s, a)]. \end{aligned}$$

□

Theorem. *The relation between LfD, naive LfO, and inverse dynamics disagreement can be characterized as*

$$\mathbb{D}_{\text{KL}} (\rho_\pi(a|s, s') || \rho_E(a|s, s')) = \mathbb{D}_{\text{KL}} (\rho_\pi(s, a) || \rho_E(s, a)) - \mathbb{D}_{\text{KL}} (\rho_\pi(s, s') || \rho_E(s, s')). \quad (13)$$

Proof. We can subtract the Kullback–Leibler divergence between the state transition of expert and agent $\mathbb{D}_{\text{KL}}(\rho_\pi(s, s') || \rho_E(s, s'))$ from the corresponding discrepancy over joint distribution $\mathbb{D}_{\text{KL}}(\rho_\pi(s, a, s') || \rho_E(s, a, s'))$ as

$$\begin{aligned} & \mathbb{D}_{\text{KL}} (\rho_\pi(s, a, s') || \rho_E(s, a, s')) - \mathbb{D}_{\text{KL}} (\rho_\pi(s, s') || \rho_E(s, s')) \\ &= \int_{\mathcal{S} \times \mathcal{A} \times \mathcal{S}} \rho_\pi(s, a, s') \left(\log \frac{\rho_\pi(s, a, s')}{\rho_E(s, a, s')} \times \frac{\rho_E(s, s')}{\rho_\pi(s, s')} \right) ds da ds' \\ &= \int_{\mathcal{S} \times \mathcal{A} \times \mathcal{S}} \rho_\pi(s, a, s') \log \frac{\rho_\pi(a|s, s')}{\rho_E(a|s, s')} ds da ds' \\ &= \mathbb{D}_{\text{KL}} (\rho_\pi(a|s, s') || \rho_E(a|s, s')). \end{aligned} \quad (14)$$

With Lemma 1, we have

$$\begin{aligned} & \mathbb{D}_{\text{KL}} (\rho_\pi(s, a, s') || \rho_E(s, a, s')) - \mathbb{D}_{\text{KL}} (\rho_\pi(s, s') || \rho_E(s, s')) \\ &= \mathbb{D}_{\text{KL}} (\rho_\pi(s, a) || \rho_E(s, a)) - \mathbb{D}_{\text{KL}} (\rho_\pi(s, s') || \rho_E(s, s')). \end{aligned} \quad (15)$$

With (14), (15)

$$\mathbb{D}_{\text{KL}} (\rho_\pi(a|s, s') || \rho_E(a|s, s')) = \mathbb{D}_{\text{KL}} (\rho_\pi(s, a) || \rho_E(s, a)) - \mathbb{D}_{\text{KL}} (\rho_\pi(s, s') || \rho_E(s, s')).$$

□

We now introduce the following lemma that will be used in the proof for Corollary 1.

Lemma 2. *if the dynamics is injective, i.e., distribution $\mathcal{T}(s'|s, a)$ that characterizes the dynamics is a degenerate distribution, and there is only one possible action corresponds to a state transition, the conditional distribution $\rho_\pi(a|s, s')$ that characterizes the corresponding inverse model under policy π will also be injective, and will be independent with policy π , thus we have*

$$\rho_\pi(a|s, s') = \rho_E(a|s, s'), \quad (16)$$

where $\rho_E(a|s, s')$ characterizes the corresponding inverse dynamics model for the expert.

Proof. We will begin with the definition of inverse model as

$$\rho_\pi(a|s, s') = \frac{\mathcal{T}(s'|s, a)\pi(a|s)}{\int_A \mathcal{T}(s'|\bar{s}, \bar{a})\pi(\bar{a}|s)d\bar{a}}. \quad (17)$$

Since \mathcal{T} is injective, which means that $\mathcal{T}(s'|s, a) = \delta(s' - f(s, a))$, $f : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$, f is a deterministic function that independent with policy π , δ is Dirac delta function. When $f(s, a) = s'$, and for given s, s' , there is only one a satisfy this equation, we have

$$\begin{aligned} \rho_\pi(a|s, s') &= \frac{\delta(0) \times \pi(a|s)}{1 \times \pi(\bar{a} = a|s)} \\ &= \delta(0). \end{aligned} \quad (18)$$

When $f(s, a) \neq s'$, it will be

$$\begin{aligned} \rho_\pi(a|s, s') &= \frac{0 \times \pi(a|s)}{\int_A \mathcal{T}(s'|\bar{s}, \bar{a})\pi(\bar{a}|s)d\bar{a}} \\ &= 0. \end{aligned} \quad (19)$$

Finally we can rewrite $\rho_\pi(a|s, s')$ as

$$\rho_\pi(a|s, s') = \begin{cases} \delta(0) & f(s, a) = s', \\ 0 & f(s, a) \neq s' \end{cases}, \quad (20)$$

which is independent with current policy π , thus we have

$$\rho_\pi(a|s, s') = \rho_E(a|s, s') = \begin{cases} \delta(0) & f(s, a) = s', \\ 0 & f(s, a) \neq s' \end{cases}. \quad (21)$$

□

Corollary. If the dynamics $\mathcal{T}(s'|s, a)$ is injective, LfD is equivalent to naive LfO.

$$\mathbb{D}_{\text{KL}}(\rho_\pi(s, a) || \rho_E(s, a)) = \mathbb{D}_{\text{KL}}(\rho_\pi(s, s') || \rho_E(s, s')). \quad (22)$$

Proof. With Lemma 1, we can substitute the right side of the equality as

$$\begin{aligned} &\mathbb{D}_{\text{KL}}(\rho_\pi(s, a) || \rho_E(s, a)) \\ &= \mathbb{D}_{\text{KL}}(\rho_\pi(s, a, s') || \rho_E(s, a, s')) \\ &= \mathbb{E}_{\rho_\pi} \left[\log \frac{\rho_\pi(s, a, s')}{\rho_\pi(s, a, s')} \right] \\ &= \mathbb{E}_{\rho_\pi} \left[\frac{\rho_\pi(s, s')\rho_\pi(a|s, s')}{\rho_E(s, s')\rho_E(a|s, s')} \right] \\ &= \underbrace{\mathbb{E}_{\rho_\pi} \left[\frac{\rho_\pi(s, s')}{\rho_E(s, s')} \right]}_{\text{by Lemma 2}} \\ &= \mathbb{D}_{\text{KL}}(\rho_\pi(s, s') || \rho_E(s, s')). \end{aligned} \quad (23)$$

□

Theorem. Let $\mathcal{H}_\pi(s, a)$ and $\mathcal{H}_E(s, a)$ denote the causal entropies over the state-action occupancy measures of the agent and expert, respectively. When $\mathbb{D}_{\text{KL}}[\rho_\pi(s, s') || \rho_E(s, s')]$ is minimized, we have

$$\mathbb{D}_{\text{KL}}[\rho_\pi(a|s, s') || \rho_E(a|s, s')] \leq -\mathcal{H}_\pi(s, a) + \text{Const.} \quad (24)$$

Proof. We will begin with the gap as the discrepancy between the inverse model of agent and expert

$$\begin{aligned} &\mathbb{D}_{\text{KL}}(\rho_\pi(a|s, s') || \rho_E(a|s, s')) \\ &= \int_{\mathcal{S} \times \mathcal{A} \times \mathcal{S}} \rho_\pi(s, a, s') \log \frac{\rho_\pi(s, a, s')\rho_E(s, s')}{\rho_E(s, a, s')\rho_\pi(s, s')} dsda ds' \end{aligned}$$

$$\begin{aligned}
&= \underbrace{\int_{\mathcal{S} \times \mathcal{A} \times \mathcal{S}} \rho_\pi(s, a, s') \log \frac{\rho_\pi(s, a) \rho_E(s, s')}{\rho_E(s, a) \rho_\pi(s, s')} dsdads'}_{\text{by Lemma 1}} \\
&= \underbrace{\int_{\mathcal{S} \times \mathcal{A} \times \mathcal{S}} \rho_\pi(s, a, s') \log \frac{\rho_\pi(s, a)}{\rho_E(s, a)} dsdads'}_{\mathbb{D}_{\text{KL}}(\rho_\pi(s, s') || \rho_E(s, s')) = 0} \\
&= -\mathcal{H}_\pi(s, a) - \int_{\mathcal{S} \times \mathcal{A}} \rho_\pi(s, a) \log \rho_E(s, a) dsda \\
&\leq -\mathcal{H}_\pi(s, a) + \sup_{\rho_\pi} \left(- \int_{\mathcal{S} \times \mathcal{A}} \rho_\pi(s, a) \log \rho_E(s, a) dsda \right) \\
&= -\mathcal{H}_\pi(s, a) + \text{Const.}
\end{aligned} \tag{25}$$

Note that the second term in the inequality $\sup_{\rho_\pi} (\cdot)$ cannot be optimized w.r.t. the parameterized policy π_θ and thus can be omitted from the objective of maximizing $\mathcal{H}_\pi(s, a)$. \square

A.2 Theorem 1 and its Corollary with Jensen-Shannon Divergence

Lemma 3 (Lemma 1 with JS divergence). *The equality below holds.*

$$\mathbb{D}_{\text{JS}}[\rho_\pi(s, a, s') || \rho_E(s, a, s')] = \mathbb{D}_{\text{JS}}[\rho_\pi(s, a) || \rho_E(s, a)]. \tag{26}$$

Proof. We can expand the left side of the equality as

$$\begin{aligned}
&\mathbb{D}_{\text{JS}}[\rho_\pi(s, a, s') || \rho_E(s, a, s')] \\
&= \mathbb{E}_{\rho_\pi} \left[\frac{1}{2} \log \frac{\rho_\pi(s, a, s')}{\frac{1}{2}\rho_\pi(s, a, s') + \frac{1}{2}\rho_E(s, a, s')} \right] + \mathbb{E}_{\rho_E} \left[\frac{1}{2} \log \frac{\rho_E(s, a, s')}{\frac{1}{2}\rho_\pi(s, a, s') + \frac{1}{2}\rho_E(s, a, s')} \right] \\
&= \mathbb{E}_{\rho_\pi} \left[\frac{1}{2} \log \frac{\rho_\pi(s, a) \mathcal{T}(s'|s, a)}{(\frac{1}{2}\rho_\pi(s, a) + \frac{1}{2}\rho_E(s, a)) \mathcal{T}(s'|s, a)} \right] \\
&\quad + \mathbb{E}_{\rho_E} \left[\frac{1}{2} \log \frac{\rho_E(s, a) \mathcal{T}(s'|s, a)}{(\frac{1}{2}\rho_\pi(s, a) + \frac{1}{2}\rho_E(s, a)) \mathcal{T}(s'|s, a)} \right] \\
&= \mathbb{E}_{\rho_\pi} \left[\frac{1}{2} \log \frac{\rho_\pi(s, a)}{\frac{1}{2}\rho_\pi(s, a) + \frac{1}{2}\rho_E(s, a)} \right] + \mathbb{E}_{\rho_E} \left[\frac{1}{2} \log \frac{\rho_E(s, a)}{\frac{1}{2}\rho_\pi(s, a) + \frac{1}{2}\rho_E(s, a)} \right] \\
&= \mathbb{D}_{\text{JS}}[\rho_\pi(s, a) || \rho_E(s, a)].
\end{aligned} \tag{27}$$

\square

Theorem 3 (Theorem 1 with JS divergence). *Given optimal expert policy π_E , $\rho_E(s, a)$, $\rho_E(s, s')$ denote its state-action and state transition occupancy measures accordingly, the optimization gap between minimizing the discrepancy of these two types of occupancy measures w.r.t. agent policy π shows that*

$$\mathbb{D}_{\text{JS}}(\rho_\pi(a|s, s') || \rho_E(a|s, s')) = \mathbb{D}_{\text{JS}}(\rho_\pi(s, a) || \rho_E(s, a)) - \mathbb{D}_{\text{JS}}(\rho_\pi(s, s') || \rho_E(s, s')) + \epsilon, \tag{28}$$

where the minor term ϵ will converge to zero when minimizing the naïve LfO objective under JS divergence $\mathbb{D}_{\text{JS}}(\rho_\pi(s, s') || \rho_E(s, s'))$.

Proof. We will begin with subtracting the Jensen-Shannon divergence between the state transition of expert and agent $\mathbb{D}_{\text{JS}}(\rho_\pi(s, s') || \rho_E(s, s'))$ from the corresponding discrepancy over joint distribution $\mathbb{D}_{\text{JS}}(\rho_\pi(s, a) || \rho_E(s, a))$ as

$$\begin{aligned}
&\mathbb{D}_{\text{JS}}(\rho_\pi(s, a) || \rho_E(s, a)) - \mathbb{D}_{\text{JS}}(\rho_\pi(s, s') || \rho_E(s, s')) \\
&= \underbrace{\mathbb{D}_{\text{JS}}(\rho_\pi(s, a, s') || \rho_E(s, a, s')) - \mathbb{D}_{\text{JS}}(\rho_\pi(s, s') || \rho_E(s, s'))}_{\text{by Lemma 3}} \\
&= \int_{\mathcal{S} \times \mathcal{A} \times \mathcal{S}} \frac{1}{2} \rho_\pi(s, a, s') \log \left(\frac{\rho_\pi(s, a, s')}{\frac{1}{2}(\rho_\pi(s, a, s') + \rho_E(s, a, s'))} \times \frac{\frac{1}{2}(\rho_\pi(s, s') + \rho_E(s, s'))}{\rho_\pi(s, s')} \right) dsdads'
\end{aligned}$$

$$\begin{aligned}
& + \int_{\mathcal{S} \times \mathcal{A} \times \mathcal{S}} \frac{1}{2} \rho_E(s, a, s') \log \left(\frac{\rho_E(s, a, s')}{\frac{1}{2}(\rho_\pi(s, a, s') + \rho_E(s, a, s'))} \times \frac{\frac{1}{2}(\rho_\pi(s, s') + \rho_E(s, s'))}{\rho_E(s, s')} \right) dsdads' \\
& = \int_{\mathcal{S} \times \mathcal{A} \times \mathcal{S}} \left(\frac{1}{2} \rho_\pi(s, a, s') (\log \rho_\pi(a|s, s') + \log f(s, a, s')) \right. \\
& \quad \left. + \frac{1}{2} \rho_E(s, a, s') (\log \rho_E(a|s, s') + \log f(s, a, s')) \right) dsdads'. \tag{29}
\end{aligned}$$

We denote (29) as $\Gamma(s, a, s')$ and $f(s, a, s') = \frac{\rho_\pi(s, s') + \rho_E(s, s')}{\rho_\pi(s, a, s') + \rho_E(s, a, s')}$. We then expand the discrepancy between the inverse model of expert and agent as

$$\begin{aligned}
& \mathbb{D}_{\text{JS}}(\rho_\pi(a|s, s') || \rho_E(a|s, s')) \\
& = \int_{\mathcal{S} \times \mathcal{A} \times \mathcal{S}} \frac{1}{2} \rho_\pi(s, a, s') \log \frac{2\rho_\pi(a|s, s')}{\rho_\pi(a|s, s') + \rho_E(a|s, s')} dsdads' \\
& \quad + \int_{\mathcal{S} \times \mathcal{A} \times \mathcal{S}} \frac{1}{2} \rho_E(s, a, s') \log \frac{2\rho_E(a|s, s')}{\rho_\pi(a|s, s') + \rho_E(a|s, s')} dsdads'.
\end{aligned}$$

When $\mathbb{D}_{\text{JS}}(\rho_\pi(s, s') || \rho_E(s, s')) \rightarrow 0$, there will be $\frac{\rho_\pi(s, s')}{\rho_E(s, s')} \rightarrow 1$, and $g(s, a, s') \rightarrow 2$. Therefore $\Gamma(s, a, s') - \mathbb{D}_{\text{JS}}(\rho_\pi(a|s, s') || \rho_E(a|s, s')) \rightarrow 0$. If we denote

$$\begin{aligned}
\epsilon &= \Gamma(s, a, s') - \mathbb{D}_{\text{JS}}(\rho_\pi(a|s, s') || \rho_E(a|s, s')) \\
&= \mathbb{D}_{\text{JS}}(\rho_\pi(s, a) || \rho_E(s, a)) - \mathbb{D}_{\text{JS}}(\rho_\pi(s, s') || \rho_E(s, s')) - \mathbb{D}_{\text{JS}}(\rho_\pi(a|s, s') || \rho_E(a|s, s')),
\end{aligned}$$

we get $\epsilon \rightarrow 0$ during the minimization of $\mathbb{D}_{\text{JS}}(\rho_\pi(s, s') || \rho_E(s, s'))$. \square

Corollary 2 (Corollary 1 with JS divergence). *If the dynamics $\mathcal{T}(s'|s, a)$ is injective, LfD is equivalent to naive LfO (replacing KL with JS divergence).*

$$\mathbb{D}_{\text{JS}}(\rho_\pi(s, a) || \rho_E(s, a)) = \mathbb{D}_{\text{JS}}(\rho_\pi(s, s') || \rho_E(s, s')). \tag{30}$$

Proof. With Lemma 3, we can substitute the right side of the equality as

$$\begin{aligned}
& \mathbb{D}_{\text{JS}}(\rho_\pi(s, a) || \rho_E(s, a)) \\
& = \mathbb{D}_{\text{JS}}(\rho_\pi(s, a, s') || \rho_E(s, a, s')) \\
& = \mathbb{E}_{\rho_\pi} \left[\frac{1}{2} \log \frac{2\rho_\pi(s, a, s')}{\rho_\pi(s, a, s') + \rho_E(s, a, s')} \right] + \mathbb{E}_{\rho_E} \left[\frac{1}{2} \log \frac{2\rho_E(s, a, s')}{\rho_\pi(s, a, s') + \rho_E(s, a, s')} \right] \\
& = \mathbb{E}_{\rho_\pi} \left[\frac{1}{2} \log \frac{2\rho_\pi(s, s') \rho_\pi(a|s, s')}{\rho_\pi(s, s') \rho_\pi(a|s, s') + \rho_E(s, s') \rho_E(a|s, s')} \right] \\
& \quad + \mathbb{E}_{\rho_E} \left[\frac{1}{2} \log \frac{2\rho_E(s, s') \rho_E(a|s, s')}{\rho_\pi(s, s') \rho_\pi(a|s, s') + \rho_E(s, s') \rho_E(a|s, s')} \right] \\
& = \underbrace{\mathbb{E}_{\rho_\pi} \left[\frac{1}{2} \log \frac{2\rho_\pi(s, s')}{\rho_\pi(s, s') + \rho_E(s, s')} \right] + \mathbb{E}_{\rho_E} \left[\frac{1}{2} \log \frac{2\rho_E(s, s')}{\rho_\pi(s, s') + \rho_E(s, s')} \right]}_{\text{by Lemma 2}} \\
& = \mathbb{D}_{\text{JS}}(\rho_\pi(s, s') || \rho_E(s, s')). \tag{31}
\end{aligned}$$

\square

A.3 Theorem 2 with Jensen-Shannon Divergence

Assumption 2. *Given the inverse dynamics model of agent $\rho_\pi(a|s, s')$ and expert $\rho_E(a|s, s')$. The following inequality*

$$\mathbb{D}_{\text{KL}}(\rho_E(a|s, s') || \rho_\pi(a|s, s')) \leq \mathbb{D}_{\text{KL}}(\rho_\pi(a|s, s') || \rho_E(a|s, s')) + \delta, \tag{32}$$

where δ is a minor term that will converge to 0 and thus can be omitted during the minimization the inverse dynamics disagreement between $\rho_\pi(a|s, s')$ and $\rho_E(a|s, s')$, should always holds, or the reverse Kullback-Leibler divergence of the inverse model between agent and expert should be bounded by the KL divergence between them.

Note that, this assumption is somewhat trivial since when KL divergence $\mathbb{D}_{\text{KL}}(\rho_\pi(a|s, s')||\rho_E(a|s, s'))$ is sufficiently minimized, the total variance between $\rho_\pi(a|s, s')$ and $\rho_E(a|s, s')$ is also minimized, thus inverse $\mathbb{D}_{\text{KL}}(\rho_E(a|s, s')||\rho_\pi(a|s, s'))$ will be minimized at the same time. Apparently, the inequality holds and there will be $\delta \rightarrow 0$.

Theorem 4 (Theorem 2 with JS divergence). *Let $\mathcal{H}_\pi(s, a)$ and $\mathcal{H}_E(s, a)$ denote the causal entropies over the state-action occupancy measures of the agent and expert, respectively. When $\mathbb{D}_{\text{KL}}[\rho_\pi(s, s')||\rho_E(s, s')]$ is minimized, we have*

$$\mathbb{D}_{\text{JS}}[\rho_\pi(a|s, s')||\rho_E(a|s, s')] \leq -\mathcal{H}_\pi(s, a) + \text{Const.} \quad (33)$$

Proof. We will begin with the gap as the discrepancy between the inverse model of agent and expert

$$\begin{aligned} & \mathbb{D}_{\text{JS}}[\rho_\pi(a|s, s')||\rho_E(a|s, s')] \\ &= \int_{\mathcal{S} \times \mathcal{A} \times \mathcal{S}} \frac{1}{2} \rho_\pi(s, a, s') \log \frac{2}{1 + \frac{\rho_E(a|s, s')}{\rho_\pi(a|s, s')}} dsdads' \\ &\quad + \int_{\mathcal{S} \times \mathcal{A} \times \mathcal{S}} \frac{1}{2} \rho_E(s, a, s') \log \frac{2}{1 + \frac{\rho_\pi(a|s, s')}{\rho_E(a|s, s')}} dsdads' \\ &= \log 2 - \int_{\mathcal{S} \times \mathcal{A} \times \mathcal{S}} \frac{1}{2} \rho_\pi(s, a, s') \log \left(1 + \frac{\rho_E(a|s, s')}{\rho_\pi(a|s, s')}\right) dsdads' \\ &\quad - \int_{\mathcal{S} \times \mathcal{A} \times \mathcal{S}} \frac{1}{2} \rho_E(s, a, s') \log \left(1 + \frac{\rho_\pi(a|s, s')}{\rho_E(a|s, s')}\right) dsdads' \\ &\leq \log 2 + \int_{\mathcal{S} \times \mathcal{A} \times \mathcal{S}} \frac{1}{2} \rho_\pi(s, a, s') \log \frac{\rho_\pi(a|s, s')}{\rho_E(a|s, s')} dsdads' \\ &\quad + \int_{\mathcal{S} \times \mathcal{A} \times \mathcal{S}} \frac{1}{2} \rho_E(s, a, s') \log \frac{\rho_E(a|s, s')}{\rho_\pi(a|s, s')} dsdads' \\ &= \frac{1}{2} \mathbb{D}_{\text{KL}}(\rho_\pi(a|s, s')||\rho_E(a|s, s')) + \frac{1}{2} \mathbb{D}_{\text{KL}}(\rho_E(a|s, s')||\rho_\pi(a|s, s')) + \log 2 \\ &= \underbrace{\mathbb{D}_{\text{KL}}(\rho_\pi(a|s, s')||\rho_E(a|s, s'))}_{\text{by Assumption 2}} + \delta + \log 2 \\ &\leq \underbrace{-\mathcal{H}_\pi(s, a) + \text{Const.}}_{\text{by Theorem 2}}, \end{aligned} \quad (34)$$

where the first inequality is given by Jensen's inequality, and as we demonstrated in Assumption 2, the minor error δ will converge to 0 and thus can be omitted from the objective of minimizing $\mathbb{D}_{\text{KL}}(\rho_\pi(a|s, s')||\rho_E(a|s, s'))$. \square

A.4 Gradient Estimation of Policy Entropy and Mutual Information

Here we provide the policy gradient formula for causal entropy $\mathcal{H}_\pi(a|s)$ and mutual information $\mathcal{I}(s; (s', a))$.

Proposition 1. *The policy gradient of causal entropy $\mathcal{H}_\pi(a|s)$ is given by*

$$\begin{aligned} \nabla_\theta \mathbb{E}_{\pi_\theta}[-\log \pi_\theta(a|s)] &= \mathbb{E}_\theta [\nabla_\theta \log \pi_\theta(a|s) Q^H(s, a)], \\ \text{where } Q^H(\bar{s}, \bar{a}) &= \mathbb{E}_{\pi_\theta}[-\log \pi_\theta(a|s)|s_0 = \bar{s}, a_0 = \bar{a}]. \end{aligned} \quad (35)$$

Proof. Define $\rho(s) = \sum_a \rho(s, a)$ as the state occupancy measure. Then we have

$$\begin{aligned} \nabla_\theta \mathbb{E}_{\pi_\theta}[-\log \pi_\theta(a|s)] &= -\nabla_\theta \sum_{s,a} \rho_{\pi_\theta}(s, a) \log \pi_\theta(a|s) \\ &= -\sum_{s,a} (\nabla_\theta \rho_{\pi_\theta}(s, a)) \log \pi_\theta(a|s) - \sum_s \rho_{\pi_\theta}(s) \sum_a \pi_\theta(a|s) \nabla_\theta \log \pi_\theta(a|s) \\ &= -\sum_{s,a} (\nabla_\theta \rho_{\pi_\theta}(s, a)) \log \pi_\theta(a|s) - \underbrace{\sum_s \rho_{\pi_\theta}(s) \sum_a \nabla_\theta \pi_\theta(a|s)}_{=0} \end{aligned}$$

$$= - \sum_{s,a} (\nabla_\theta \rho_{\pi_\theta}(s,a)) \log \pi_\theta(a|s). \quad (36)$$

This is exactly the policy gradient for RL with fixed cost function $c(s,a) = \log \pi_\theta(a|s)$. And the resulting policy gradient (35) is given by the standard policy gradient with cost $c(s,a)$. \square

Proposition 2. *The policy gradient of mutual information $\mathcal{I}(s; (s', a))$ is given by*

$$\begin{aligned} \nabla_\theta \mathcal{I}_{\pi_\theta}(s; (s', a)) &= \mathbb{E}_\theta [\nabla_\theta \log \pi_\theta(a|s) Q^I(s, a)], \\ \text{where } Q^I(\bar{s}, \bar{a}) &= \mathcal{I}_{\pi_\theta}(s; (s', a)|s_0 = \bar{s}, a_0 = \bar{a}). \end{aligned} \quad (37)$$

Proof. Note that $\mathcal{I}_\pi(s; (s', a)) = \mathcal{H}_\pi(s) - \mathcal{H}_\pi(s|s', a)$. The same as the proof for Proposition 1, we have

$$\begin{aligned} \nabla_\theta \mathcal{I}_{\pi_\theta}(s; (s', a)) &= -\nabla_\theta \sum_s \rho_{\pi_\theta}(s) \log \pi_\theta(s) - \nabla_\theta \sum_{s,a,s'} \rho_{\pi_\theta}(s, a, s') \log \pi_\theta(s|s', a) \\ &= -\sum_s (\nabla_\theta \rho_{\pi_\theta}(s)) \log \pi_\theta(s) - \sum_{s,a,s'} (\nabla_\theta \rho_{\pi_\theta}(s, a, s')) \log \pi_\theta(s|s', a). \end{aligned} \quad (38)$$

We can thus see $\mathcal{I}_\pi(s; (s', a))$ as a fixed cumulative cost sum of a MDP, thus the resulting policy gradient will be (37). \square

B Specifications

B.1 Hyperparameters

Tab. 3 lists the parameters for BCO [42], DeepMimic [28], GAIL [16], GAIfO [43] and proposed method used in the comparative evaluation.

Table 3: Hyperparameters for Evaluated Algorithms

Parameter	Value
<i>Shared</i>	
Optimizer	Adam [21]
Learning rate	$3e^{-4}$
Batch size	512
Discount (γ)	0.99
Architecture of policy, value and discriminator networks	(300, 400)
Nonlinearity	Tanh
<i>BCO</i>	
Inverse model training epoches	50
<i>DeepMimic</i>	
Reward type	Joint angle q_t and Joint velocity \dot{q}_t
Reward design	$r_t = w_p \exp(-2(\sum_j \ \hat{q}_t^j - q_t^j\ _2)) + w_v \exp(-0.1(\sum_j \ \hat{\dot{q}}_t^j - \dot{q}_t^j\ _2))$
Reward weight (w_p, w_v)	(0.8, 0.2)
<i>GAIL</i>	
Weight of policy entropy	0.01
<i>Ours</i>	
Weight of policy entropy (λ_p)	0.01
Weight of state entropy (λ_s)	0.1
Pretrained MI estimator steps	10000
Update MI estimator steps	50
Architecture of MI estimator network	(512, 512)

B.2 Gridworld Environment and Inverse Dynamics Disagreement

We will first demonstrate how our Gridworld environments are motivated by illustrating the relation between inverse dynamics disagreement and possible functional-equivalent action choices. Then we will provide the detailed specifications of our Gridworld environment.

The intuition behind the design of these experiments is that the complexity of the dynamics shows positive correlation with inverse dynamics disagreement. Under the deterministic dynamics, the complexity will be mainly dominated by the numbers of action choices (or size of state space, but we override it as we adopt a fixed size maze). Consider a MDP with two state s_0, s_1 and a set of actions $\{a_{01}^{(0)}, a_{01}^{(1)}, a_{01}^{(2)}, \dots\}$ that can let the agent transform from s_0 to s_1 . To approximately compute inverse dynamics disagreement, we denote $\pi(a|s=s_0) \sim \text{Categorical}(p_1 = p_2 = \dots = p_k)$ is a uniformed initialized policy on a discrete action space with size k , $\pi_\theta(a|s=s_0)$ is a θ -parameterized expert policy. Without loss of generality, we assume θ has a prior of normal distribution $\theta \sim \mathcal{N}(\mathbf{0}^k, \mathbf{1}^k)$. Therefore, we can approximately compute inverse dynamics disagreement as follows.

$$\begin{aligned} & \text{Inverse Dynamics Disagreement} \\ & \approx \mathbb{E}_{\theta \sim \mathcal{N}(\mathbf{0}^k, \mathbf{1}^k)} [\mathbb{D}_{\text{KL}} (\pi(a|s=s_0)p(s=s_0) || \pi_\theta(a|s=s_0)p(s=s_0))] , \end{aligned} \quad (39)$$

where $p(s = s_0)$ is the distribution of state. Since there is only two states available, $p(s = s_0) = 1$. Fig. 1a in the main paper are plotted with (39). As we can see, inverse dynamics disagreement does show a growing trend as the number of possible action choices increases.

To this end, we design several simple Gridworld environments (see Fig. 1) to help understand how inverse dynamics disagreement affects the imitation learning algorithms. The red block is the starting point of the agent, while the agent is encouraged to move toward the target green block. All the black and dark grey block are permitted to move through, while the grey block represents wall. The action that the agent may conduct including four basic ones: *moving left*, *moving right*, *moving up*, *moving down* when the number of possible action choices is one. If the number of possible action choices is larger than one (*e.g.* n choices), there will be $n - 1$ functional equivalent choices added to each original moving action, *i.e.*, now there will be n action choices for moving left/right/up/down. For the reward strategy, once the agent successfully reaches the green target block, it will receive a reward of 100, and the game will immediately come to an end. When the agent takes an original moving action, it will receive a penalty of -1 , but when the agent chooses an action choice that is other than the original one, it will receive a penalty of -5 . All the numerical evaluation results are under this strategy.

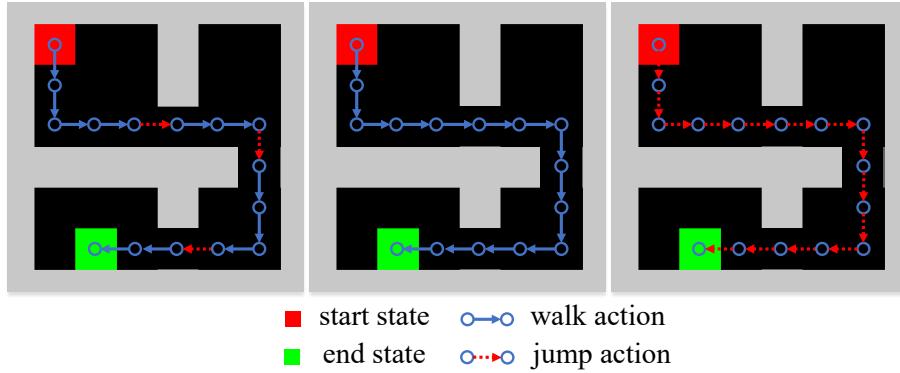


Figure 6: Gridworld environment.

B.3 Other Environments

Tab. 4 lists the specifications about the benchmark environments and number of state transition pairs (state-action pairs for GAIL) in demonstration for each environment.

Table 4: Specifications for Evaluated Environments

Environment	\mathcal{S}	\mathcal{A}	Max-Step	Demonstration Size
CartPole	\mathbb{R}^4	$\{0, 1\}$	200	5000
Pendulum	\mathbb{R}^4	\mathbb{R}^1	1000	50000
DoublePendulum	\mathbb{R}^{11}	\mathbb{R}^1	1000	50000
Hopper	\mathbb{R}^{11}	\mathbb{R}^3	1000	50000
Halfcheetah	\mathbb{R}^{17}	\mathbb{R}^6	1000	50000
Ant	\mathbb{R}^{111}	\mathbb{R}^8	1000	50000

C Additional Empirical Results

C.1 Quantitative Results of Toy Example

Tab. 5 lists the quantitative results of the toy Gridworld experiments.

Table 5: Quantitative Results of GAIL, GAIfO and our method in Gridworld Environment.

Num. of Action	1	2	4	11
GAIL [16]	86.0 \pm 3.0	70.4 \pm 6.4	68.7 \pm 5.8	69 \pm 4.0
GAIfO [43]	86.8 \pm 1.3	55.7 \pm 11.9	48.3 \pm 9.3	28.3 \pm 6.2
Ours	87.3 \pm 1.8	65.0 \pm 3.3	56.0 \pm 5.0	49.0 \pm 8.6

C.2 Comparative Evaluations

On the differences on results compared with [43] For the baseline results of GAIfO [43], we notice that there are some differences between the results reported in [43] and our paper (Tab. 2 and Fig. 2 in the main paper). We hypothesis that the reason is twofold. First, **different physics engine**. Referring to the footnote 2 in page 5 of [44], the experiments in [43] are conducted with PyBullet [8] physics engine, while we use MuJoCo [41] instead since it is the default physics engine in OpenAI Gym [7] benchmark. Second, **different expert demonstrations**. As [43] does not provide the expert demonstrations used for imitation learning, we collect the demonstrations for all the baselines and our method by training an expert with PPO [34], which may lead to different imitation learning results.

C.3 Quantitative Results of Ablation Study

Tab. 6 and Tab. 7 list the quantitative results of the ablations analysis (sensitivity to policy entropy and mutual information), while the corresponding learning curves can be found in Fig. 7a and Fig. 7b respectively.

Table 6: Quantitative results about λ_p on *HalfCheetah* task.

hyperparameters	Averaged return
$\lambda_p = 0.0, \lambda_s = 0.01$	4882.8 \pm 40.1
$\lambda_p = 0.0005, \lambda_s = 0.01$	5526.2 \pm 95.6
$\lambda_p = 0.001, \lambda_s = 0.01$	5343.2 \pm 88.5
$\lambda_p = 0.01, \lambda_s = 0.01$	5404.8 \pm 103.7

Table 7: Quantitative results about λ_s on *HalfCheetah* task.

hyperparameters	Averaged return
$\lambda_p = 0.001, \lambda_s = 0.0$	4658.0 \pm 90.2
$\lambda_p = 0.001, \lambda_s = 0.001$	5189.7 \pm 77.2
$\lambda_p = 0.001, \lambda_s = 0.01$	5343.2 \pm 88.5
$\lambda_p = 0.001, \lambda_s = 0.1$	5540.5 \pm 100.3

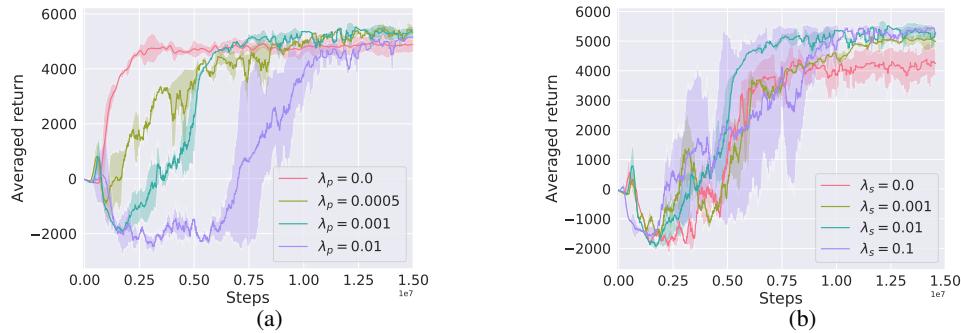


Figure 7: (a) Learning curves of our method under different λ_p settings. (b) Learning curves of our method under different λ_s settings.

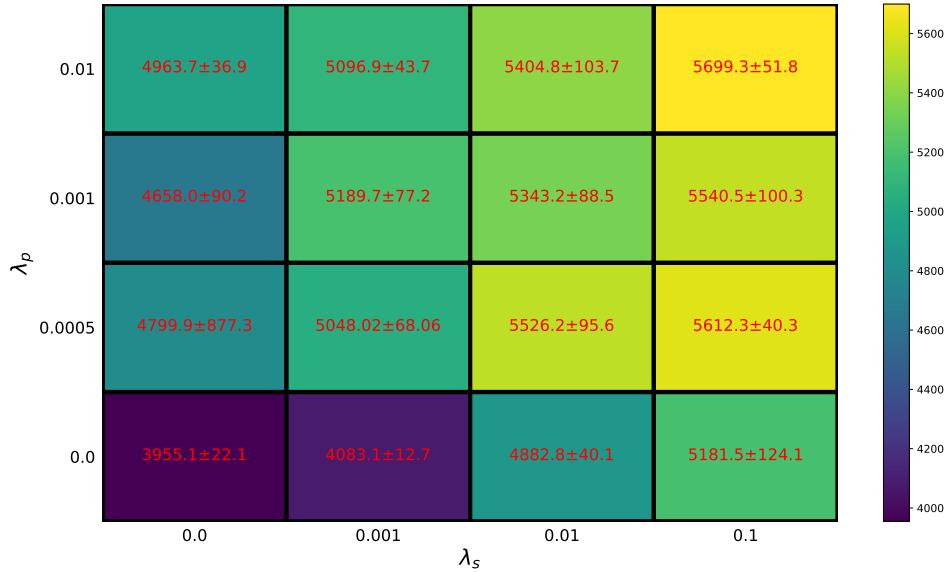


Figure 8: Quantitative results of a grid search on λ_s and λ_p .

To further illustrate how our method can benefit from the two components (policy entropy and MI terms), here we also provide the results of performing a grid search on λ_p and λ_s in Fig. 8. All the numerical results are evaluated under the same criteria as other experiments.

The results read that, adding MI term can always promote the imitation performances, and the improvement can be more significant as the value of λ_s increases. And the promotions it obtains are robust to the changes of λ_p . On the other hand, imitation performance can also benefit from adding policy entropy, while different λ_p may lead to different improvements over the GAIfO baseline (the left-bottom block, with $\lambda_s = \lambda_p = 0$).