Expert Proximity as Surrogate Rewards for Single Demonstration Imitation Learning

Chia-Cheng Chiang [*1] **Li-Cheng Lan** [*2] **Wei-Fang Sun** [3] **Chien Feng** [1] **Cho-Jui Hsieh** [2] **Chun-Yi Lee** [1]

Abstract

In this paper, we focus on single-demonstration imitation learning (IL), a practical approach for real-world applications where acquiring multiple expert demonstrations is costly or infeasible and the ground truth reward function is not available. In contrast to typical IL settings with multiple demonstrations, single-demonstration IL involves an agent having access to only one expert trajectory. We highlight the issue of sparse reward signals in this setting and propose to mitigate this issue through our proposed Transition Discriminator-based IL (TDIL) method. TDIL is an IRL method designed to address reward sparsity by introducing a denser surrogate reward function that considers environmental dynamics. This surrogate reward function encourages the agent to navigate towards states that are proximal to expert states. In practice, TDIL trains a transition discriminator to differentiate between valid and non-valid transitions in a given environment to compute the surrogate rewards. The experiments demonstrate that TDIL outperforms existing IL approaches and achieves expert-level performance in the single-demonstration IL setting across five widely adopted MuJoCo benchmarks as well as the "Adroit Door" robotic environment.

1. Introduction

Single-demonstration imitation learning (or simply "*single-demo IL*" hereafter) is characterized by an agent having access to only one expert demonstration (i.e., a single expert trajectory). This contrasts with typical IL settings, where multiple demonstrations are available (Ho & Ermon, 2016; Fu et al., 2020). Both settings allow interactions with the environment during training but lack access to the ground truth reward function, online human feedback, or prior knowledge acquired from analogous tasks. Single-demo IL is a practical paradigm for addressing real-world challenges, as collecting a large number of expert demonstrations is often expensive and sometimes not even feasible, especially in applications such as autonomous robots. Consider the training of a surgical robot (Ou & Tavakoli, 2023). In situations where certain surgical procedures are extremely rare, it is possible that only a single expert surgeon's demonstration is available. Similarly, in the context of training an agent for vehicle control (Scheel et al., 2022), unique scenarios such as stabilizing the vehicle during a tire blowout, navigating icy roads, or avoiding collisions with objects may have only one or very few demonstrations available. Another example is cooking tutorials on YouTube, where YouTubers typically demonstrate the cooking process only once. As a result, a robotic agent learning from a single-demonstration setting encounters similar challenges in these domains. However, many IL methods, such as behavior cloning (BC) and most basic inverse reinforcement learning (IRL) methods, face limitations when only a single demonstration is available. BC tends to overfit when few expert demonstrations are provided. For basic IRL methods, the scarcity of expert demonstrations can typically result in a sparse reward situation, which may lead to relatively limited training signals for the agent. This issue of reward sparsity becomes even more pronounced in high-dimensional, continuous environments with randomly initialized positions. In light of these, developing an effective and robust learning mechanism that operates solely with a single demonstration is of considerable importance. Unfortunately, although a few previous methods (Dadashi et al., 2021; Freund et al., 2023) exist that can be utilized to address few-demonstration IL, single demo IL remains relatively unexplored and offers opportunities for further advancing contemporary IL approaches.

To confront the single-demo IL paradigm, this study proposes an IRL method with a denser reward function, termed Transition Discriminator-based IL (TDIL). TDIL increases the density of obtainable reward signals in the IRL setting while accounting for environmental dynamics to ensure rea-

[*]Equal contribution [1]ELSA Lab, Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan. [2]Department of Computer Science, University of California, Los Angeles, CA, USA. [3]NVIDIA AI Technology Center, NVIDIA Corporation, Santa Clara, CA, USA. Correspondence to: Chun-Yi Lee <cylee@cs.nthu.edu.tw>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

sonable agent behavior. A motivational example illustrating this concept is provided in Fig. 1 (a). If an agent finds itself in a cell that allows for a direct transition to an expert state (e.g., the green arrows in Fig. 1 (a)), the most reasonable action is to facilitate this transition. Based on this concept, TDIL derives a surrogate reward function that rewards the agent for moving toward states close to expert states. Although there exist other dense reward IL methods (e.g., PWIL (Dadashi et al., 2021) and FISH (Haldar et al., 2023b)) designed to guide the agent to states that are close to expert states, the distance metrics they employed, such as the Euclidean (L2) or the cosine distance metrics, are not theoretically sound. For example, in Fig. 1 (a), if the L2 distance is adopted, two adjacent grid cells would be considered close to each other even if a barrier exists between them. This could potentially lead the agent to cells that are either infeasible or unable to reach expert states in an efficient manner. In contrast, TDIL takes environmental dynamics into consideration by leveraging a well-trained *transition discriminator*, which adopts a training objective aimed at distinguishing between valid and non-valid transitions regarding two states' reachability in a given environment. As a result, TDIL is able to construct a more reasonable and denser reward function (e.g., Fig. 1 (d)) for guiding the agent back to expert states in the single-demo IL setting.

To validate the efficacy of TDIL, we perform comprehensive experiments on five widely adopted MuJoCo benchmarks (Todorov et al., 2012), aligning with most prior IL research, as well as the "Adroit Door" environment (Rajeswaran et al., 2017) in the Gymnasium-Robotics collection (de Lazcano et al., 2023). The experimental evidence reveals that TDIL delivers exceptional performance, matches expert-level results on these benchmarks, and outperforms existing IL approaches. Moreover, another key insight from our experiments is the significant correlation between the derived reward signals and the inaccessible ground truth reward signals. This correlation offers a practical solution for blind model selection by selecting a checkpoint without the help of the ground truth reward function. This differentiates TDIL from prior work that relied on environment rewards at test time for early termination or optimal model selection, which are assumptions that are impractical in the general IL context. The main contributions are summarized as follows:

1. We highlight the limitations of previous IL methods under the challenging single-demo IL setting. These methods may produce sparse reward signals or sometimes even overlook the dynamics of the environment.
2. We introduce a novel TDIL algorithm, which utilizes a dense and dynamics-aware surrogate reward function.
3. We validate that our surrogate reward function is effective for blind model selection scenarios without requiring access to the ground truth reward function.

2. Preliminary

Reinforcement learning (RL). An MDP is typically formalized as a tuple $\langle \mathcal{S}, \mathcal{A}, P, R, p_0 \rangle$, where $\mathcal{S}$ represents the state space, $\mathcal{A}$ the action space, $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ the transition function, $R(s, a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ the reward function, and $p_0(s_0)$ the distribution of the initial state s_0. The transition function $P(s_{t+1}|s_t, a_t)$ specifies the probability of transitioning to state s_{t+1} upon taking action a_t in state s_t. Within this MDP, a trajectory τ is defined as a sequence of states and actions $[s_0, a_0, s_1, a_1, \ldots, s_T, a_T]$, where s_0 is sampled from the distribution p_0, and s_{t+1} is the resulting state after taking action a_t in state s_t. The objective of an RL policy $\pi(a|s, \theta)$ is to learn a set of parameters θ that maximizes the expected total return $\mathbb{E}_{\tau \sim p(\tau|\theta)} \sum_{t=0}^{T} R(s_t, a_t)$.
Single-demo IL. During the training of single-demo IL settings, the agent can interact with the environment. However, it does not have access to the reward function R. Instead, the agent is given a trajectory $\tau_e = [s_0^e, a_0^e, s_1^e, a_1^e, \ldots, s_T^e, a_T^e]$ generated by an expert policy π_e in the same environment as a hint of the reward R. As a result, the goal of single-demo IL is to train a policy that can converge to the expert demonstration even when initiated from a different initial state s_0, and faithfully follow the expert actions when within the support of the demonstration. After training, the performance of it is evaluated by the ground truth reward function R.

Inverse reinforcement learning (IRL). IRL methods constitute a type of IL that aims to learn or infer the reward function based on provided demonstrations. (Levine, 2018) demonstrated that the objective of IRL is to learn a Conditional Probability Distribution (CPD) denoted as $p(\mathcal{O}_t = 1|s_t, a_t)$. In this expression, the optimal indicator $\mathcal{O}_t$ serves as a binary random variable that indicates whether the time step t is optimal. Specifically, in the context of IRL, $\mathcal{O}_t = 1$ if the (s_t, a_t) pair is present in an expert trajectory. Furthermore, the CPD $p(\mathcal{O}_t = 1|s_t, a_t)$ can be marginalized to form $p(\mathcal{O}_t = 1|s_t) = \int_{\mathcal{A}} p(a_t|s_t) p(\mathcal{O}_t = 1|s_t, a_t) da_t$. By assuming the action prior $p(a|s)$ produces the expert actions in the expert states, $p(\mathcal{O}_t = 1|s_t) = 1$ if and only if s_t is an expert state. The assumption can be ensured through BC or GAIL. In the following sections, we slightly abuse notation (i.e., dropping $= 1$) as in Levine (2018) for the sake of conciseness by expressing $p(\mathcal{O} = 1)$ as $p(\mathcal{O})$.

3. Analysis on the Sparse Reward Issue

To examine the sparse reward issue in single-demo IL scenarios, we design a 2D grid-world environment and compare the reward functions, learned optimal policies, and training curves across different IL methods. Fig. 1 (a) illustrates the grid-world environment, where the circled triangle symbol denotes the initial state of the expert, while the red lines represent barriers that obstruct certain paths. The blue arrows trace the path of the expert's demonstration as it progresses

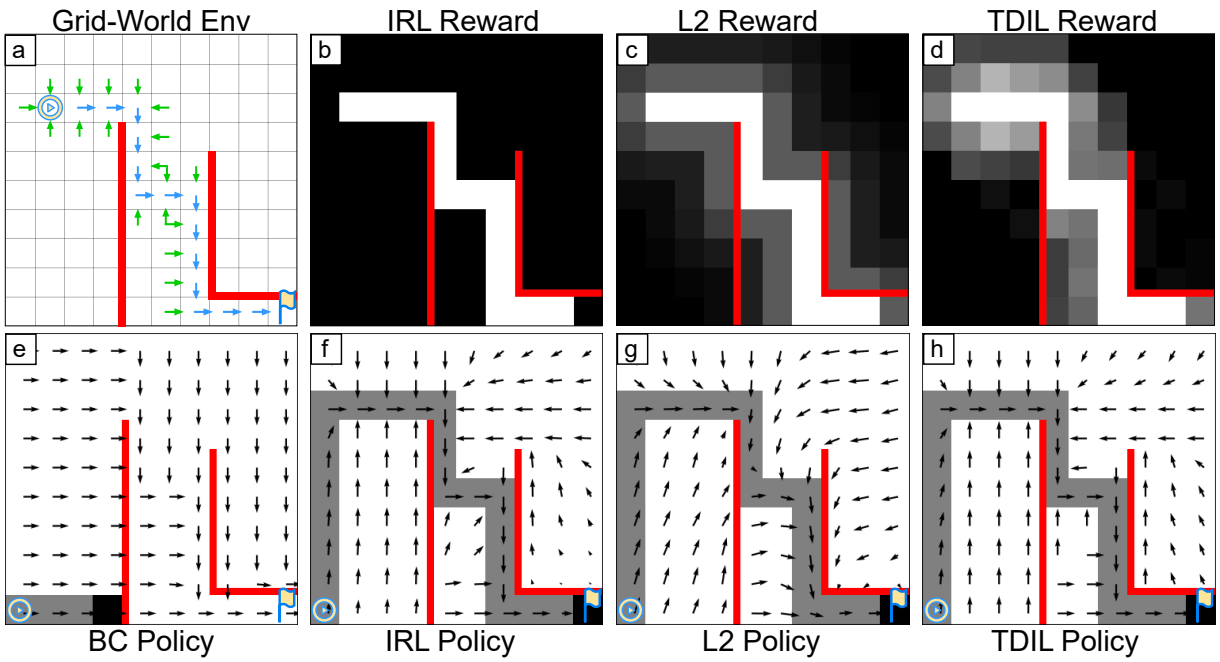


Figure 1. A motivational grid-world example for comparing different IL methods trained with the single-demo IL setting. (a) depicts the expert's demonstration, denoted by blue arrows, while red lines represent impassable barriers, reflecting environmental dynamics. The green arrows symbolize the state-action pairs that are one step directed toward the expert states. Subfigures (b)-(d) present reward signals calculated through various methods: (b) using the basic IRL method (i.e., GAIL (Ho & Ermon, 2016)), (c) based on the L2 distance between the agent's and the expert's state-action pairs, and (d) through our proposed TDIL. Finally, subfigures (e)-(h) illustrate the actions calculated by averaging the directions represented by the logits for the discrete actions from the learned policy at distinct grid locations.

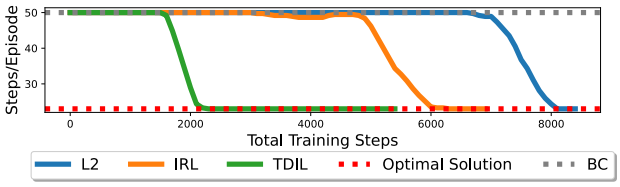


Figure 2. The total steps per episode required for each agent to reach the goal in the grid-world, with a fixed limit of 50 steps. Lower values for "Steps per episode" indicate better efficiency.

toward the goal state, which is marked by a yellow flag.

We investigate three distinct approaches for defining the reward functions: (1) the basic IRL reward (i.e., Ho & Ermon (2016)), (2) the L2 distance reward, and (3) the TDIL reward. The basic IRL reward only provides rewards to the agent when it performs expert demonstrations (i.e., the blue arrows). The L2 distance reward provides rewards according to the L2 distance between the current state-action of the agent and that of the expert. This reward represents methods that use geometric distance to measure the spatial disparity between the two (e.g, PWIL (Dadashi et al., 2021), FISH (Haldar et al., 2023b), and ROT (Haldar et al., 2023a)). Finally, our TDIL reward considers the transitions to expert state reasonable (i.e., the green arrows) and provides rewards to them. The corresponding reward functions of these three cases are visualized in Figs. 1 (b)-(d). Subsequently, we train Soft Actor-Critic (SAC) (Haarnoja et al., 2018) agents under a uniform initial state distribution $p_0(s_0)$, using each of these reward functions. The critic in SAC facilitates the propagation of reward signals to cells that do not provide any rewards. In Figs. 1 (f)-(h), the agents all start at the bottom-left corner, with the learned policy at each state indicated by an arrow and the trajectories of the agents highlighted with a grey background. The policy learned through BC is included in Fig. 1 (e) for comparison. The training curves are presented in Fig. 2, indicating the steps required to reach the goal from the bottom-left corner.

According to the above setup, it can be observed that in this single-demo IL scenario, the basic IRL method results in a sparse reward function (i.e., Fig. 1 (b)), and necessitates more training steps to converge, as illustrated by the orange curve in Fig. 2. This issue becomes more pronounced in high-dimensional environments with a continuous state space, where the expert trajectory may represent a low-dimensional manifold with measure zero. In such scenarios, basic IRL methods that aimed to minimize the f-divergences between the agent and expert state distributions often encounter convergence challenges. These challenges stem from the difficulties of matching two manifolds with significantly different dimensions (Arjovsky & Bottou, 2017).

For the L2 distance reward, although it leads to a more densely defined reward function as illustrated in Fig. 1 (c),

the learned policy tends to become trapped in the states above the goal state, which are obstructed by red barriers, as depicted in Fig. 1 (g). This can be attributed to the reliance on distance measures that do not adequately reflect the state transition dynamics inherent in the underlying MDP. Despite the proximity to the expert support based on the L2 distance, these states are actually far separated when the influence of state transition dynamics is taken into account. Although this issue can be alleviated by a better state representation, tailoring such representations for different environments can be a challenging task. This process may require domain-specific knowledge and may not be generalized well to unseen environments. Moreover, even with a perfect representation, geometric distances such as L2 and cosine similarity may not adequately capture the dynamics of certain environments. Consider the example of driving a car on a highway. Geometric distances treat moving forward and backward as equally valid options, which is reflected in the symmetry property ($L2(f(s_i), f(s_j)) = L2(f(s_j), f(s_i))$, where f is the representation function). However, in reality, car movement on a highway is asymmetric. While it is possible to move forward freely, attempting to move backward at high speeds is dangerous or even impossible. As a result, using geometric distance to define a dense reward function may not suit environments with complex transition functions.

Finally, the proposed TDIL method results in a denser surrogate reward function, while considering the state transition dynamics of the MDP, as shown in Fig. 1 (d). TDIL achieves this by providing rewards to the state-action pairs that return to expert proximity states (including states highlighted by yellow in Fig. A1). While these state-action pairs may not necessarily correspond to the shortest path to the goal state, we posit that returning to the expert support first is the most conservative decision for out-of-distribution states (i.e., Fig. 1 (h)), given the absence of the ground truth reward function. Moreover, introducing incentives for the agent to navigate to the expert support can lead to an accelerated convergence speed, as indicated by the green curve in Fig. 2.

By employing TDIL to define a denser surrogate reward function, the sparse reward issue is mitigated when compared to using the basic IRL methods. In high-dimensional environments, even when provided with an expert trajectory within a low-dimensional manifold with measure zero, TDIL can offer the potential to define a manifold with a higher dimension and a non-zero measure, thereby improving the stability of IRL methods in matching the agent and expert state distributions. On the other hand, in comparison to the methods that use geometry distance, TDIL exhibits the ability to assign more reasonable rewards due to its awareness of state transition dynamics, as evidenced by the states above the goal state in Figs. 1 (c)-(d). This feature prevents the agent from being trapped in states with high L2 distance rewards that are distant from the expert support. As a result, TDIL holds the potential to enhance training efficiency by adopting a dense and dynamics-aware surrogate reward function, which enables the agent to propagate reward signals back across the states in the environment.

4. Methodology

Section 4.1 introduces the expert reachability indicator to establish the concept of expert proximity. Subsequently, a denser, dynamics-aware surrogate reward function is formally defined. However, the computation of surrogate rewards is intractable. To address this challenge, Section 4.2 presents an approximation for the surrogate reward function. Section 4.3 describes the steps to realize this approximation through the use of a transition discriminator. Building upon these concepts, Section 4.4, introduces a practical algorithm named TDIL, which is designed to facilitate concurrent training of the transition discriminator and the agent. Moreover, Section 4.4 discusses the advantage of using state based transition discriminator over state-action based one. Finally, Section 4.6 explores the capability of using TDIL for blind model selection, which enables the selection of a proper checkpoint without relying on ground truth rewards.

4.1. Expert Proximity and Surrogate Rewards

Based on the optimal indicator $\mathcal{O}_t$ described in Section 2, we define the expert reachability indicator $\tilde{\mathcal{O}}_t$, which identifies the state-action pairs capable of returning to an expert state (i.e., green arrows in Fig. 1 (a)). For a given state-action pair (s_t, a_t), we define $p(\tilde{\mathcal{O}}_t|s_t, a_t)$ based on $\mathcal{O}_t$, with $p(\tilde{\mathcal{O}}_t|s_t, a_t)$ indicating the probability of reaching an expert state by selecting action a_t in state s_t. Taking Fig. 1 (a) as an example, if (s_t, a_t) is one of the green or blue arrows, $p(\tilde{\mathcal{O}}_t|s_t, a_t) = 1$ as the agent can reach an expert state from state s_t. Formally, we define $p(\tilde{\mathcal{O}}_t|s_t, a_t)$ as follows:

$$p(\tilde{\mathcal{O}}_t|s_t, a_t) \stackrel{\text{def}}{=} \int_{\mathcal{S}} P(s_{t+1}|s_t, a_t) p(\mathcal{O}_{t+1}|s_{t+1}) ds_{t+1}, \tag{1}$$

where P is the state transition function of the MDP and $p(\mathcal{O}_{t+1}|s_{t+1})$ is the probability of s_{t+1} being an expert state. Given the action prior $p(a|s)$ described in Section 2, $p(\tilde{\mathcal{O}}_t|s_t)$ can be derived by marginalizing $p(\tilde{\mathcal{O}}_t|s_t, a_t)$ as:

$$\begin{aligned} p(\tilde{\mathcal{O}}_t|s_t) &= \int_{\mathcal{A}} p(a_t|s_t) p(\tilde{\mathcal{O}}_t|s_t, a_t) da_t \\ &= \int_{\mathcal{A}} p(a_t|s_t) \int_{\mathcal{S}} P(s_{t+1}|s_t, a_t) p(\mathcal{O}_{t+1}|s_{t+1}) ds_{t+1} da_t \\ &= \int_{\mathcal{S}} p(\mathcal{O}_{t+1}|s_{t+1}) \int_{\mathcal{A}} p(a_t|s_t) P(s_{t+1}|s_t, a_t) da_t ds_{t+1}. \end{aligned} \tag{2}$$

Based on $p(\tilde{\mathcal{O}}_t|s_t)$, we define *expert proximity* as the set of states capable of transitioning to expert states within a single action. In other words, a state s_t is in expert proximity if

and only if $\tilde{\mathcal{O}}_t = 1$. Nevertheless, calculating $p(\mathcal{O}_{t+1}|s_{t+1})$ requires access to the ground truth expert support, which is unavailable in general. Fortunately, in the context of single-demonstration IL settings, it is possible to derive the probability $\hat{p}(\tilde{\mathcal{O}}_t|s_t)$ of reaching expert states as follows:

$$\begin{aligned}\hat{p}(\tilde{\mathcal{O}}_t|s_t) &= \sum_{i=0}^{N} p(\mathcal{O}_{t+1}|s_i^e) \int_{\mathcal{A}} p(a_t|s_t) P(s_i^e|s_t, a_t) da_t \\ &= \sum_{i=0}^{N} \int_{\mathcal{A}} p(a_t|s_t) P(s_i^e|s_t, a_t) da_t,\end{aligned} \tag{3}$$

where s_i^e denotes the i-th state in the expert demonstration, and N represents the total number of expert states. Finally, we define our surrogate reward function $R_{\text{TDIL}}(s_t, a_t)$ as:

$$R_{\text{TDIL}}(s_t, a_t) \stackrel{\text{def}}{=} \mathbb{E}_{s_{t+1} \sim P(s_{t+1}|s_t, a_t)} \Big[\hat{p}(\tilde{\mathcal{O}}_{t+1}|s_{t+1}) \Big], \tag{4}$$

which assigns positive surrogate rewards when transitioning to states in expert proximity. This denser reward function also facilitates the propagation of rewards to earlier states in the agent's trajectory and, therefore, can potentially improve its training speed and efficiency as discussed in Section 3.

4.2. Approximating the Surrogate Reward

The computation of R_{TDIL} involves an intractable integration term $\int_{\mathcal{A}} p(a_t|s_t) P(s_i^e|s_t, a_t) da_t$ as specified in Eq. (3). To circumvent the complexity introduced by this intractable term, we assume that the action prior $p(a_t|s_t)$ is optimal and deterministic in the states that are in expert proximity. This enables us to reformulate the intractable term as follows:

$$\int_{\mathcal{A}} p(a_t|s_t) P(s_i^e|s_t, a_t) da_t = \max_a P(s_i^e|s_t, a). \tag{5}$$

Eq. (5) determines an agent's capability to transition from s_t to s_i^e, which cannot be computed directly due to the inaccessibility of state transition dynamics. As a result, we train a transition discriminator $D_\phi(s_i, s_j)$ to approximate the state transition dynamics, which determines whether a given state s_i can reach another state s_j within a single timestep. For example, for any tuple (s_t, a_t, s_{t+1}) in the replay buffer, $D_\phi(s_t, s_{t+1})$ should return 1 since the tuple evidences the reachability. The optimal transition discriminator $D_{\phi^*}(s_i, s_j)$ can be formally defined as follows:

$$D_{\phi^*}(s_i, s_j) \stackrel{\text{def}}{=} \max_{a_i} \mathbb{1}[P(s_j|s_i, a_i) > 0]. \tag{6}$$

The surrogate rewards R_{TDIL} can then be approximated as:

$$R_{\text{TDIL}}(s_i, s_j) \approx \mathbb{E}_{s_{t+1} \sim p(s_{t+1}|s_t, a_t)} \Big[\sum_{i=0}^{N} D_\phi(s_{t+1}, s_i^e) \Big], \tag{7}$$

where the workflow of approximating R_{TDIL} through the use of a given transition discriminator D_ϕ is depicted in Fig. 3.

4.3. Training the Transition Discriminator

To train D_ϕ, we optimize it using maximum likelihood training, with the binary cross-entropy loss L_D defined as:

$$\begin{aligned}L_D = -\Big(&\alpha \mathbb{E}_{(s_i, s_j) \sim B^+} \big[\log \left(D_\phi(s_i, s_j) \right) \big] + \\ &(1-\alpha) \mathbb{E}_{(s_i, s_j) \sim B^-} \big[\log \left(1 - D_\phi(s_i, s_j) \right) \big] \Big),\end{aligned} \tag{8}$$

where $\alpha \in (0, 1)$ is a balancing coefficient, B^+ is the set of positive samples, and B^- is the set of negative samples. In practice, we choose the set of positive samples B^+ as:

$$B^+ = \{(s, s') \mid (s, a, s') \in B\}, \tag{9}$$

where B is the replay buffer. For negative samples B^-, we choose the union of the set of contrastive samples (i.e., easy negative samples) $B^-_{\text{contrastive}}$ and the set of reversed transition samples (i.e., hard negative samples) B^-_{reversed} as:

$$\begin{aligned}&B^- = B^-_{\text{contrastive}} \cup B^-_{\text{reversed}}, \text{where} \\ &B^-_{\text{contrastive}} = \{(s_i, s_j) \mid (s_i, a_i, s_{i+1}), (s_j, a_j, s_{j+1}) \in B\}, \\ &B^-_{\text{reversed}} = \{(s', s) \mid (s, a, s') \in B\}.\end{aligned} \tag{10}$$

The positive samples are taken from valid transitions collected by the agent. For the negative samples, we assume that two randomly sampled states seldom represent a valid transition and that the majority of reversed transitions are likely to be invalid. Based on this assumption, the transition discriminator is trained with millions of positive and negative state transitions gathered through the agent's interaction with the environment during training. This method mitigates the likelihood of overfitting compared to the previous work (Ho & Ermon, 2016), which uses only expert demonstrations as positive examples for training the discriminator.

4.4. The TDIL Algorithm

Fig. 4 presents an overview of the proposed TDIL algorithm, which involves repeating the following four steps. First, the agent interacts with the environment and stores the collected transitions in the replay buffer. These transitions are then utilized to update the transition discriminator according to Eq. (8) in the second step. In the third step, a batch of transitions is sampled from the replay buffer to calculate the aggregated reward R_{agg}, which is defined as the following:

$$R_{\text{agg}}(s_t, a_t) \stackrel{\text{def}}{=} \beta R_{\text{IRL}}(s_t, a_t) + (1-\beta) R_{\text{TDIL}}(s_t, a_t), \tag{11}$$

where β is a hyperparameter for balancing between the two rewards. The aggregated rewards R_{agg} combine the basic IRL rewards R_{IRL}, which ensures optimality on the expert and expert proximity states, and the proposed R_{TDIL}, which incentivizes the agent to navigate towards states that are in expert proximity. In the fourth step, the sampled transitions and the aggregated rewards are utilized to train a

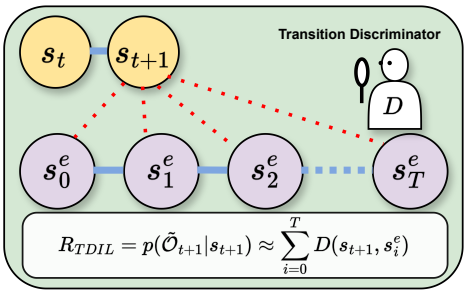


Figure 3. The approximation of R_{TDIL} through the use of a pretrained transition discriminator D.

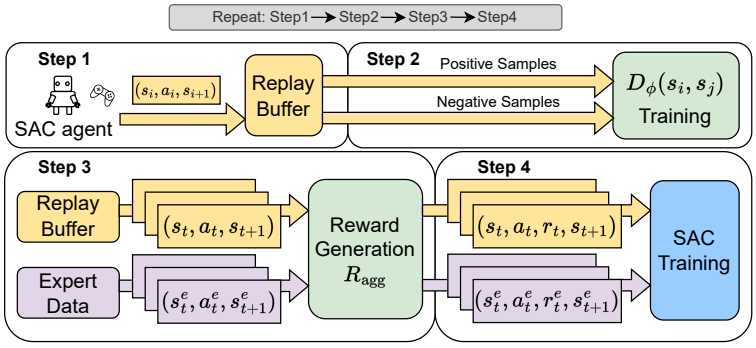


Figure 4. An overview of the TDIL method. **Step 1:** Agent-environment interaction. **Step 2:** Transition discriminator updates. **Step 3:** Generation of aggregated rewards. **Step 4:** Training an RL agent based on the generated reward signals.

SAC agent. This iterative process facilitates the concurrent training of the transition discriminator and the agent. For the pseudocode and additional details of the proposed TDIL algorithm, please refer to Section A.2. In practice, the basic IRL rewards R_{IRL} are calculated using GAIL. In addition, an ablation study that explores different choices of β is presented in Section A.4.9. Furthermore, we explore another variant where R_{IRL} (i.e., $\beta = 0$) is removed, and an additional BC loss is employed to train the policy to ensure the optimality on expert states, as described in Section 2.

4.5. Distinctions of State and State-Action based TDs

While a state-action discriminator could potentially increase the density of reward signals, our TDIL reward is designed based on a state discriminator for the following two reasons. First, the state-based discriminator provides rewards on a larger number of state-action pairs, which can result in denser rewards compared to a state-action based discriminator. For a transition (s_t, a_t, s_{t+1}), the state discriminator provides rewards as long as s_{t+1} is in the expert proximity. In contrast, a state-action based discriminator would provide rewards only when s_t is in the expert proximity. Furthermore, training a state-action discriminator $D'(s_i, a_i, s_j)$ can be more challenging since it not only requires ensuring that s_t can transition to s_j but also necessitates validating whether a_i is the permissible action for such a transition.

4.6. Relative Rewards for Blind Model Selection

In practice, we find that the normalized surrogate rewards can effectively select a decent model from a collection of training checkpoints, without the need for direct access to ground truth rewards R. This attribute is noteworthy in the context of IL applications, where the best model may not be the one trained for the longest, as detailed in Appendix A.4.6. This capability is realized through the computation of relative total rewards $\sum_{t=0}^{T} R_{\text{TDIL}}(s_t, a_t) / \sum_{i=0}^{N} R_{\text{TDIL}}(s_i^e, a_i^e)$, instead of using the raw return $\sum_{t=0}^{T} R_{\text{TDIL}}(s_t, a_t)$. These relative total rewards serve as a decent indicator for selecting the best-performing model. Note that the implementation details of the relative returns are provided in Appendix A.4.5.

5. Experimental Results

This section presents our experimental results conducted in two distinct environments: MuJoCo (Todorov et al., 2012) and Adroit Hand (Rajeswaran et al., 2017). We also include ablation studies to provide deeper insights into our method.

5.1. Baselines

We have selected BC (Bain & Sammut, 1995), GAIL (Ho & Ermon, 2016), f-IRL (Ni et al., 2020), PWIL (Dadashi et al., 2021), and CFIL (Freund et al., 2023) as our baseline methods. GAIL is a widely recognized and extensively adopted IL method. CFIL and f-IRL both represent the state-of-the-art (SOTA) adversarial-based methods, while PWIL serves as a representative of non-adversarial approaches. IQ-Learn (Garg et al., 2021) is excluded from our comparison due to its subpar performance (Zeng et al., 2022; Sikchi et al., 2022). The original f-IRL study did not include evaluations on *Humanoid-v3*. Our assessments revealed that its performance on *Humanoid-v3* was considerably below acceptable levels. LS-IQ (Al-Hafez et al., 2023) is also excluded due to the unavailability of its complete code. We do not include AIRL (Fu et al., 2017) since the previous study (Ni et al., 2020) has demonstrated that AIRL exhibits inferior performance compared to f-IRL (Ni et al., 2020), particularly under settings with few expert demonstrations.

5.2. Single-Demo IL Evaluation Results and Insights

In this section, we compare two variants of TDIL described in Section 4.4, with baseline methods under the single-demo IL setting. The first variant, denoted as Ours (R_{TDIL} + BC), replaces R_{IRL} with BC loss. In this variant, we set $\beta = 0$ in R_{agg} to assess whether our surrogate reward R_{TDIL} alone can effectively guide the agent back to expert states. Given that

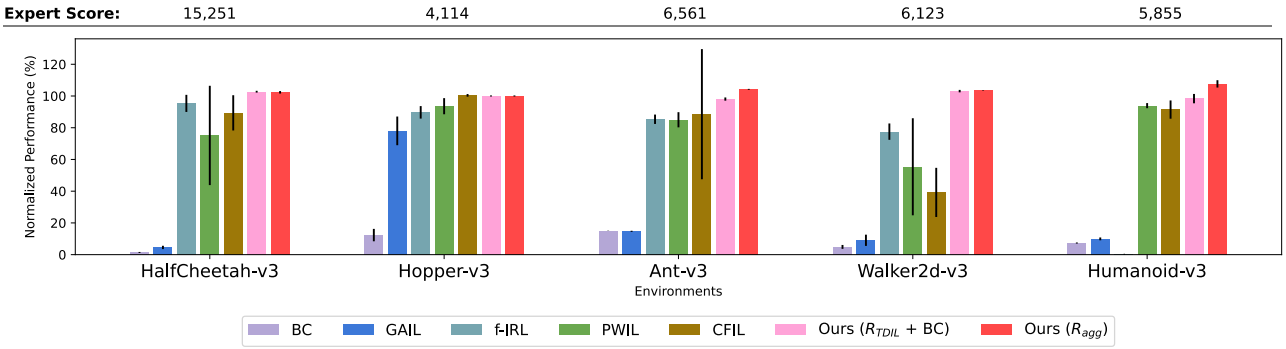


Figure 5. Normalized performance evaluation of different methodologies using the Oracle model selection under the single-demo setting.

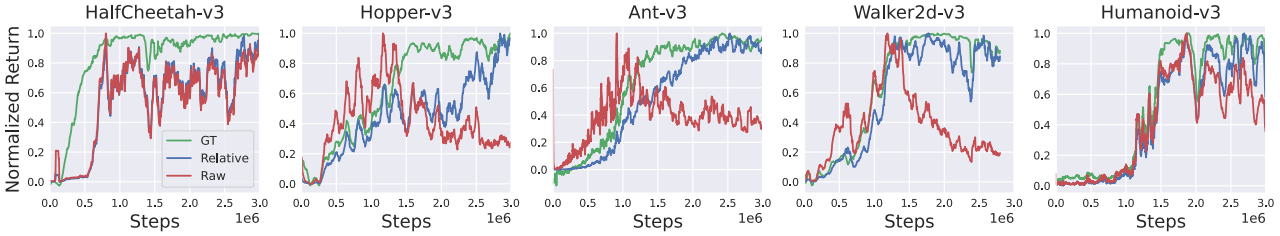


Figure 6. Comparison of normalized ground truth, raw, and relative return for blind model selection.

R_{IRL} is not utilized in this variant, BC is applied to ensure that the policy learns the expert demonstration. The second variant, denoted as Ours (R_{agg}), explores the potential of training SAC agents by R_{agg}, without the help of BC loss.

We first conduct experiments in five MuJoCo environments, including *HalfCheetah-v3*, *Hopper-v3*, *Ant-v3*, *Humanoid-v3*, and *Walker2d-v3*. All algorithms are trained over 3M timesteps, each employing five random seeds. The expert demonstrations are generated by a well-trained SAC (Haarnoja et al., 2018) agent using the default parameters. Note that the expert demonstrations we collected have higher total returns than those used by some of the baseline papers, which might lead to different results. For the aggregate reward R_{agg} version, we select $\beta = 0.9$ based on our grid search, with the results presented in Table A7.

Fig. 5 presents the evaluation results on MuJoCo, with the bars representing the normalized performance of each algorithm compared to the expert's performance. Table A2 presents the detailed testing results. Both versions of our method, i.e., Ours (R_{TDIL} + BC) and Ours (R_{agg}), achieve expert-level performance across all tested MuJoCo environments. The performance of Ours (R_{TDIL} + BC) indicates that R_{TDIL} can effectively guide agents back to the expert states, even without the use of R_{IRL}. Meanwhile, Ours (R_{agg}) demonstrates that our proposed reward function, R_{agg}, effectively serves as a reward mechanism for training SAC agents. In addition to our methods, we evaluated and compared our variants with the baselines. BC does not achieve expert-level performance in any environment, due to overfitting and its inability to generalize to out-of-distribution states. GAIL does not reach expert-level performance either, potentially due to adversarial training instability, sparse rewards in the Maximum Entropy IRL framework in single-demo IL settings, and the algorithm's low sampling efficiency. f-IRL underperforms across various tasks and is particularly ineffective on *Humanoid-v3*. PWIL does not reach expert-level performance, possibly because its reliance on Euclidean distance fails to capture the environmental dynamics correctly. This issue arises especially when two state-action pairs are close in Euclidean distance but unreachable in the MDP, which leads to inaccuracies in the computation of primal Wasserstein cost, as discussed in Section 3. Lastly, CFIL achieves expert-level performance only on *Hopper-v3*; nevertheless, it shows limited adaptability on *Walker2d-v3*.

Besides MuJoCo, we also evaluate TDIL in the Adroit Hand Door environment, as depicted in Fig. 7. The experimental settings and the complete results are detailed in Appendix A.4.4. According to the results, while two of the top baselines f-IRL and CFIL show success rates within 40% (mostly 0%), both versions of TDIL (i.e., Ours (R_{TDIL} + BC) and Ours (R_{agg})) achieve 100% success rates. This demonstrates the generalizability and robustness of TDIL.

Figure 7. The Adroit Hand Door environment illustration.

5.3. Blind Model Selection

To validate the concept discussed in Section 4.6 that relative return can be used in blind model selection, we graph

the ground truth return, raw return, as well as relative return obtained by the agent in an episode during training in Fig. 6. These rewards are all normalized by their respective maximum values over 3M timesteps. From these plots, a clear positive correlation between the relative return and the ground truth return can be observed across all environments, whereas the trends of the raw return obtained by the agents do not consistently align with the ground truth return. In particular, in *Hopper-v3*, *Ant-v3*, and *Walker2d-v3*, the raw return exhibit a trend that initially rises and then falls over 3M timesteps, a pattern that does not mirror the ground truth rewards. This fluctuation suggests that the accuracy of the transition discriminator grows during training, as it is trained on more data near the expert support collected by the increasingly proficient RL agent. Additional results on blind model selection are available in Appendix A.4.6.

5.4. Ablation Studies

This section examines the effect or performance of several components within TDIL in the Mujoco environments.

Different β in R_{agg}. β represents the weight of R_{IRL} in R_{agg}. The detailed results are reported in Table A7, which reveals that by setting β within the range of [0.1, 0.9], the agent consistently achieves expert-level performance without the use of BC loss. This highlights the adaptability and robustness of TDIL, even when certain components, such as β, are not fine-tuned for specific experimental contexts.

Training with Pure R_{TDIL}. To evaluate the performance of using R_{TDIL} solely, we present the experimental results on MuJoCo in the "w/o BC" column in Table 1. This version exhibits performance inferior to both TDIL variants: Ours (R_{TDIL} + BC) and Ours (R_{agg}). The decrease in performance highlights the significance of learning expert actions on expert states, which can be realized through the use of R_{IRL} or BC loss. However, it still achieves comparable performance against all the baselines. For instance, it attains expert-level performance in *Walker2d-v3*, where all the baselines fail.

The accuracy of the transition discriminator. To ensure the transition discriminators employed are well-trained, we report their accuracy in Table 2. The experimental results indicate that the transition discriminators can achieve an accuracy of 0.988 or higher across all environments, regardless of the dataset type (B^+, $B^-_{\text{contrastive}}$, or B^-_{reversed}). This demonstrates that our rewards are derived from well-trained transition discriminators, which can serve as trustworthy approximations employed in Eq. (7). Moreover, we evaluate the accuracy of the transition discriminators when trained with different α values in Eq. (8). The results in Table A6 reveal that the accuracy of a transition discriminator is not sensitive to the selection of hyperparameter α. This demonstrates the robustness of our proposed methodology.

Table 1. TDIL w/ and w/o the BC loss and hard negative samples.

	R_{TDIL} + BC	w/o BC	w/o B^-_{reversed}
HalfCheetah-v3	15,666 $\pm$ 85	12,630 $\pm$ 6,854	15,718 $\pm$ 179
Hopper-v3	4,115 $\pm$ 14	3,890 $\pm$ 562	4,143 $\pm$ 6
Ant-v3	6,434 $\pm$ 66	3,995 $\pm$ 2,408	6,571 $\pm$ 116
Humanoid-v3	5,758 $\pm$ 173	5,575$\pm$196	4,868$\pm$2,443
Walker2d-v3	6,312 $\pm$ 47	6,281 $\pm$ 76	6,268 $\pm$ 53

Table 2. Accuracy of the transition discriminators.

	B^+	$B^-_{\text{contrastive}}$	B^-_{reversed}
HalfCheetah-v3	1.0	0.992	0.996
Hopper-v3	1.0	0.996	0.996
Ant-v3	1.0	0.992	0.992
Humanoid-v3	1.0	0.99	0.988
Walker2d-v3	1.0	0.992	0.988

Training without B^-_{reversed}. The column labeled "w/o B^-_{reversed}" in Table 1 illustrates the impact of excluding B^-_{reversed} during training. This configuration yields comparable performance across all environments, with the exception of *Humanoid-v3*. This finding implies that in less complex environments, the information contained in hard negative samples may not be crucial. However, in the *Humanoid-v3* environment, the absence of hard negative samples adversely affects performance. This discrepancy may be attributed to the vast state space of *Humanoid-v3*, which diminishes the likelihood that easy negative samples encapsulate the essential information contained in hard negative ones. Furthermore, as *Humanoid-v3* is a more complicated environment, the agent might be sensitive to inaccurately estimated rewards resulting from the absence of hard negative samples.

Multiple expert demonstrations. Our methodology is not limited to a single demonstration setting. To validate this, we conduct additional experiments with multiple demonstrations and present the results in Table A3. These experimental results demonstrate that our method can achieve expert-level performance with additional demonstrations.

6. Related Work

IL with adversarial training. Distribution matching methods with a min-max formulation (Ho & Ermon, 2016; Fu et al., 2018; Ke et al., 2021; Ghasemipour et al., 2020; Ni et al., 2020; Swamy et al., 2021; Kostrikov et al., 2020; Camacho et al., 2021; Freund et al., 2023; K. et al., 2019; Han et al., 2022; Zeng et al., 2022; Viano et al., 2022) might induce potential instability and sub-optimality in situations with sparse demonstration data, which could compromise the effectiveness and reliability of these methodologies.

IL with support estimation. Methods that rely on expert

support estimation (Wang et al., 2019; Brantley et al., 2020; Liu et al., 2020; Kim et al., 2020) often face difficulties when expert data are limited. This is attributable to their reliance on the availability and quality of expert demonstrations, leaving them ill-suited for scenarios with scarce expert data.

IL with optimal transport. IL approaches that utilize optimal transport technique (Dadashi et al., 2021; Xiao et al.), on the other hand, are also less suitable for the single-demonstration IL setting, as they tend to overlook environmental dynamics. Specifically, these approaches might identify certain states as being close or similar based on their geometric distances of the state space or some state representations (Haldar et al., 2023a), even though these states may not be permissible for transition in a Markov Decision Process (MDP). This limitation impairs their capacity to capture the complexity and variability of environments.

IL with meta-demonstrations. In *one-shot IL* (Duan et al., 2017; Finn et al., 2017; Yu et al., 2018b; Dasari & Gupta, 2021; Yu et al., 2018a; Mandi et al., 2022; Huang et al., 2019; Netanyahu et al., 2022; Valassakis et al., 2022; Hu et al., 2020), researchers have explored the use of meta-demonstrations, which are demonstrations associated with other tasks, as a tool for pre-training before proceeding to one-shot adaptation. However, gathering a substantial volume of meta-demonstrations, which are necessary for training meta parameters prior to their one-shot utilization, can be infeasible due to the expensive nature of expert demonstrations. Note that these studies are orthogonal to our work.

IL with Ground Truth Reward Function. Some previous studies (Aytar et al., 2018; Wu et al., 2021; Peng et al., 2018) focus on improving the RL agent with the help of a single demonstration. However, they still allow their RL agent to access the ground truth reward function of the environment.

The majority of the aforementioned methods either struggle to achieve expert-level performance in high-dimensional environments or are less adept at achieving robust generalization (Ni et al., 2020; Freund et al., 2023; Dadashi et al., 2021; Al-Hafez et al., 2023). These constraints highlight the necessity for enhanced strategies in the single-demonstration IL setting. The key objectives include accommodating limited expert data while taking into account environmental dynamics. Please refer to Appendix A.1 for more details.

7. Conclusions and Future Works

In this paper, we proposed TDIL as a robust approach to address the challenges inherent in single-demo IL settings. By considering the transitions towards expert states as reasonable, we defined a dense surrogate reward function that can be approximated by a transition discriminator. Our experiments on the MuJoCo benchmarks and the Adroit Hand Door task revealed that our method consistently achieves expert-level performance and outperforms all the baseline algorithms, including BC, GAIL, f-IRL, PWIL, and CFIL. To further validate our surrogate reward function, we compared the ground truth return, raw return, and relative return, and revealed a strong correlation among them. This correlation substantiates the efficacy of our surrogate reward function for blind model selection. Furthermore, we conducted a series of ablation studies to validate the design choices behind TDIL. This work not only provides valuable insights but also lays a solid groundwork for future exploration in single-demo IL settings.

To accommodate more complex or higher-dimensional environments, a promising future direction involves extending our surrogate rewards from one-step to multi-step transitions, as briefly described in Appendix A.5. This would enable the surrogate reward function to provide rewards for a broader range of transitions and guide the agent back to the expert state more efficiently.

Acknowledgment

The authors gratefully acknowledge the support from the National Science and Technology Council (NSTC) in Taiwan under grant number MOST 111-2223-E-007-004-MY3, the financial support from MediaTek Inc., Taiwan, and the support from the National Science Foundation (NSF) under grant numbers 2048280, 2331966, 2325121, and 2244760, as well as from the Office of Naval Research (ONR) under grant number N00014-23-1-2300:P00001. The authors would also like to express their appreciation for the donation of the GPUs from NVIDIA Corporation and NVIDIA AI Technology Center (NVAITC) used in this study. Moreover, the authors extend their gratitude to the National Center for High-Performance Computing (NCHC) in Taiwan for providing the necessary computational and storage resources.

Impact Statement

This research focuses on single-demo IL and presents potential advancements for robotic and RL agent training. It aims to improve the training stability and efficiency for RL agents. It relies on publicly available data/environments, which ensures transparency and avoids the use of sensitive or proprietary information. Our method eliminates the complexity and time required for designing delicate rewards and collecting a tremendous amount of expert data. This elimination leads to more intuitive and efficient robot deployment. Specifically, our method enables the learning of complex tasks using only a single expert demonstration and can potentially enhance productivity and adaptability. Our approach has the potential to positively impact society by making robotic systems more accessible and efficient.

References

Al-Hafez, F., Tateo, D., Arenz, O., Zhao, G., and Peters, J. LS-IQ: Implicit reward regularization for inverse reinforcement learning. In Proc. of Int. Conf. on Learning Representations (ICLR), 2023. URL https://openreview.net/forum?id=o3Q4m8jg4BR.

Arjovsky, M. and Bottou, L. Towards principled methods for training generative adversarial networks. Int. Con. on Learning Representations (ICLR), 2017.

Aytar, Y., Pfaff, T., Budden, D., Paine, T., Wang, Z., and De Freitas, N. Playing hard exploration games by watching youtube. Advances in neural information processing systems, 31, 2018.

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

Brantley, K., Sun, W., and Henaff, M. Disagreement-regularized imitation learning. In Proc. Int. of Conf. on Learning Representations (ICLR), 2020. URL https://openreview.net/forum?id=rkgbYyHtwB.

Burda, Y., Edwards, H., Storkey, A., and Klimov, O. Exploration by random network distillation. In Proc. Int. of Conf. on Learning Representations (ICLR), 2019. URL https://openreview.net/forum?id=H1lJJnR5Ym.

Camacho, A., Gur, I., Moczulski, M. L., Nachum, O., and Faust, A. Sparsedice: Imitation learning for temporally sparse data via regularization. In ICML 2021 Workshop on Unsupervised Reinforcement Learning, 2021. URL https://openreview.net/forum?id=GEd7SmSpRR0.

Ciosek, K. Imitation learning by reinforcement learning. In Proc. Int. of Conf. on Learning Representations (ICLR), 2022. URL https://openreview.net/forum?id=1zwleytEpYx.

Dadashi, R., Hussenot, L., Geist, M., and Pietquin, O. Primal wasserstein imitation learning. In Proc. of Int. Conf. on Learning Representations (ICLR), 2021. URL https://openreview.net/forum?id=TtYSU29zgR.

Dasari, S. and Gupta, A. Transformers for one-shot visual imitation. In Proc. Int. of Conf. on on Robot Learning (ICRL), pp. 2071–2084. PMLR, 2021.

de Lazcano, R., Andreas, K., Tai, J. J., Lee, S. R., and Terry, J. Gymnasium robotics, 2023. URL http://github.com/Farama-Foundation/Gymnasium-Robotics.

Duan, Y., Andrychowicz, M., Stadie, B., Ho, J., Schneider, J., Sutskever, I., Abbeel, P., and Zaremba, W. One-shot imitation learning. In Proc. Int. of Conf. on Neural Information Processing Systems (NeurIPS), volume 30, 2017.

Finn, C., Yu, T., Zhang, T., Abbeel, P., and Levine, S. One-shot visual imitation learning via meta-learning. In Proc. Int. of Conf. on Robot Learning (ICRL), pp. 357–368. PMLR, 2017.

Freund, G., Sarafian, E., and Kraus, S. A coupled flow approach to imitation learning. In Proc. Int. of Conf. on Machine Learning (ICML), 2023.

Fu, J., Luo, K., and Levine, S. Learning robust rewards with adversarial inverse reinforcement learning. arXiv preprint arXiv:1710.11248, 2017.

Fu, J., Luo, K., and Levine, S. Learning robust rewards with adverserial inverse reinforcement learning. In Proc. Int. of Conf. on Learning Representations (ICLR), 2018. URL https://openreview.net/forum?id=rkHywl-sjA-sj.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4rl: Datasets for deep data-driven reinforcement learning, 2020.

Garg, D., Chakraborty, S., Cundy, C., Song, J., and Ermon, S. Iq-learn: Inverse soft-q learning for imitation. In Proc. of Int. Conf. on Neural Information Processing Systems (NeurIPS), 2021. URL https://openreview.net/forum?id=Aeo-sjxqtb5p.

Germain, M., Gregor, K., Murray, I., and Larochelle, H. Made: Masked autoencoder for distribution estimation. In Proc. of Int. Conf. on Machine Learning (ICML), 2015.

Ghasemipour, S. K. S., Zemel, R., and Gu, S. A divergence minimization perspective on imitation learning methods. In Kaelbling, L. P., Kragic, D., and Sugiura, K. (eds.), Proc. Int. of Conf. on Robot Learning (CoRL), volume 100 of Proceedings of Machine Learning Research, pp. 1259–1277. PMLR, 30 Oct–01 Nov 2020.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. Communications of the ACM, 63(11):139–144, 2020.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Proc. of Int. Conf. on Machine Learning (ICML), pp. 1861–1870. PMLR, 2018.

Haldar, S., Mathur, V., Yarats, D., and Pinto, L. Watch and match: Supercharging imitation with regularized optimal transport. In Con. on Robot Learning (CoRL), pp. 32–43. PMLR, 2023a.

Haldar, S., Pari, J., Rai, A., and Pinto, L. Teach a robot to fish: Versatile imitation from one minute of demonstrations. Proc. of Robotics: Science and Systems (RSS), 2023b.

Han, D.-S., Kim, H., Lee, H., Ryu, J., and Zhang, B.-T. Robust imitation via mirror descent inverse reinforcement learning. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), Proc. Int. of Conf. on Neural Information Processing Systems (NeurIPS), 2022. URL https://openreview.net/forum?id=huT1G2dtSr.

Ho, J. and Ermon, S. Generative adversarial imitation learning. In Proc. Int. of Conf. on Neural Information Processing Systems (NeurIPS), volume 29, 2016.

Hu, Z., Gan, Z., Li, W., Wen, J. Z., Zhou, D., and Wang, X. Two-stage model-agnostic meta-learning with noise mechanism for one-shot imitation. IEEE Access, 8: 182720–182730, 2020.

Huang, D.-A., Xu, D., Zhu, Y., Garg, A., Savarese, S., Fei-Fei, L., and Niebles, J. C. Continuous relaxation of symbolic planner for one-shot imitation learning. In Proc. Int. of Conf. on Intelligent Robots and Systems (IROS), 2019.

K., I., Agrawal, K. K., Dwibedi, D., Levine, S., and Tompson, J. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. In Proc. Int. of Conf. on Learning Representations (ICLR), 2019. URL https://openreview.net/forum?id=Hk4fpoA5Km.

Ke, L., Choudhury, S., Barnes, M., Sun, W., Lee, G., and Srinivasa, S. Imitation learning as f-divergence minimization. In In International Workshop on the Algorithmic Foundations of Robotics, pp. 313–329. Springer, 2021.

Kim, K., Jindal, A., Song, Y., Song, J., Sui, Y., and Ermon, S. Imitation with neural density models, 2020.

Kostrikov, I., Nachum, O., and Tompson, J. Imitation learning via off-policy distribution matching. In Proc. Int. of Conf. on Learning Representations (ICLR), 2020. URL https://openreview.net/forum?id=Hyg-s$_j$JC4FDr.

Levine, S. Reinforcement learning and control as probabilistic inference: Tutorial and review. 2018.

Liu, M., He, T., Xu, M., and Zhang, W. Energy-based imitation learning. In Proc. of Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS), 2020.

Mandi, Z., Liu, F., Lee, K., and Abbeel, P. Towards more generalizable one-shot visual imitation learning. In Proc. Int. of Conf. on Robotics and Automation (ICRA), 2022.

Nachum, O., Chow, Y., Dai, B., and Li, L. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. In Proc. Int. of Conf. on Neural Information Processing Systems (NeurIPS), volume 32, 2019.

Netanyahu, A., Shu, T., Tenenbaum, J., and Agrawal, P. Discovering generalizable spatial goal representations via graph-based active reward learning. In Proc. Int. of Conf. on Machine Learning (ICML), pp. 16480–16495, 2022.

Ni, T., Sikchi, H., Wang, Y., Gupta, T., Lee, L., and Eysenbach, B. f-irl: Inverse reinforcement learning via state marginal matching. In Proc. Int. of Conf. on Robot Learning, 2020.

Ou, Y. and Tavakoli, M. Towards safe and efficient reinforcement learning for surgical robots using real-time human supervision and demonstration. In 2023 International Symposium on Medical Robotics (ISMR), pp. 1–7. IEEE, 2023.

Peng, X. B., Abbeel, P., Levine, S., and Van de Panne, M. Deepmimic: Example-guided deep reinforcement learning of physics-based character skills. ACM Transactions On Graphics (TOG), 37(4):1–14, 2018.

Pomerleau, D. A. Efficient training of artificial neural networks for autonomous navigation. Neural computation, 3(1):88–97, 1991.

Rajeswaran, A., Kumar, V., Gupta, A., Vezzani, G., Schulman, J., Todorov, E., and Levine, S. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. Int. Con. on Robotics and Automation (ICRA), 2017.

Reddy, S., Dragan, A. D., and Levine, S. Sqil: Imitation learning via reinforcement learning with sparse rewards. In Proc. Int. of Conf. on Learning Representations (ICLR), 2019.

Scheel, O., Bergamini, L., Wolczyk, M., Osiński, B., and Ondruska, P. Urban driver: Learning to drive from real-world demonstrations using policy gradients. In Con. on Robot Learning (CoRL), pp. 718–728. PMLR, 2022.

Sikchi, H., Saran, A., Goo, W., and Niekum, S. A ranking game for imitation learning. Tran. on Machine Learning Research (TMLR), 2022.

Song, Y. and Kingma, D. P. How to train your energy-based models, 2021.

Sun, M., Devlin, S., Hofmann, K., and Whiteson, S. Deterministic and discriminative imitation (d2-imitation): Revisiting adversarial imitation for sample efficiency. In Proc. of the AAAI Conference on Artificial Intelligence, 2022.

Swamy, G., Choudhury, S., Bagnell, J. A., and Wu, S. Of moments and matching: A game-theoretic framework for closing the imitation gap. In Meila, M. and Zhang, T. (eds.), Proc. Int. of Conf. on Machine Learning (ICML), volume 139 of Proceedings of Machine Learning Research, pp. 10022–10032. PMLR, 18–24 Jul 2021.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In 2012 IEEE/RSJ Int. Con. on Intelligent Robots and Systems (IROS), pp. 5026–5033. IEEE, 2012. doi: 10.1109/IROS.2012.6386109.

Valassakis, E., Papagiannis, G., Di Palo, N., and Johns, E. Demonstrate once, imitate immediately (dome): Learning visual servoing for one-shot imitation learning. In 2Proc. Int. of Conf. on Intelligent Robots and Systems (IROS), pp. 8614–8621. IEEE, 2022.

Viano, L., Kamoutsi, A., Neu, G., Krawczuk, I., and Cevher, V. Proximal point imitation learning. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), Proc. Int. of Conf. on Neural Information Processing Systems (NeurIPS), 2022. URL https://openreview.net/forum?id=4iEoOIQ7nL.

Wang, R., Ciliberto, C., Amadori, P. V., and Demiris, Y. Random expert distillation: Imitation learning via expert policy support estimation. In Chaudhuri, K. and Salakhutdinov, R. (eds.), Proc. Int. of Conf. on Machine Learning (ICML), volume 97 of Proceedings of Machine Learning Research, pp. 6536–6544. PMLR, 09–15 Jun 2019.

Wu, Z., Lian, W., Unhelkar, V., Tomizuka, M., and Schaal, S. Learning dense rewards for contact-rich manipulation tasks. In 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 6214–6221. IEEE, 2021.

Xiao, H., Herman, M., Wagner, J., Ziesche, S., Etesami, J., and Linh, T. H. Wasserstein adversarial imitation learning.

Yu, T., Abbeel, P., Levine, S., and Finn, C. One-shot hierarchical imitation learning of compound visuomotor tasks. Proc. Int. of Conf. on Intelligent Robots and Systems (IROS), 2018a.

Yu, T., Finn, C., Xie, A., Dasari, S., Zhang, T., Abbeel, P., and Levine, S. One-shot imitation from observing humans via domain-adaptive meta-learning. Proc. of Robotics: Science and Systems, 2018b.

Zeng, S., Li, C., Garcia, A., and Hong, M. Maximum-likelihood inverse reinforcement learning with finite-time guarantees. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), Proc. Int. of Conf. on Neural Information Processing Systems (NeurIPS), 2022. URL https://openreview.net/forum?id=zbt3VmTsRIj.

A. Appendix

In this appendix, we provide review of related works, detailed training configurations, additional experimental results, and discussions on the proposed TDIL method. In Section A.1, a detailed review of previous works is provided. In Section A.2, we provide the training detail of the proposed TDIL algorithm. In Section A.3, we elaborate on the experimental setups as well as the model architecture adopted in our method. In Section A.4, we present additional experimental results to validate the effectiveness of our method. In Section A.5, we extend the expert reachability indicator $\tilde{\mathcal{O}}_t$ to multiple timesteps.

A.1. Extended Review of Related Work

The single-demonstration IL setting presents a unique and challenging problem domain. Earlier online IL research commonly treats this setting as a component of their ablation studies, and often overlooks its significance. This section engages in a discussion about several well-known online IL algorithms, which can be broadly grouped into two categories: adversarial-based and non-adversarial-based methods. This discussion allows us to highlight our differences from these prior methods, and delve into the priminary reasons behind their limited effectiveness in the single-demonstration IL context.

Adversarial-based methods: Adversarial-based approaches aim to align the agent's state, state-action, or state-next-state distributions with those of the expert by employing various divergence or distance measures. For instance, GAIL (Ho & Ermon, 2016) adopts the GAN-like framework (Goodfellow et al., 2020) to train a discriminator and minimize Jensen-Shannon divergence. AIRL (Fu et al., 2018), on the other hand, utilizes forward KL-divergence to derive stationary rewards and enhance transfer learning. Building upon these approaches, the authors of (Ke et al., 2021) and f-MAX (Ghasemipour et al., 2020) unify GAIL and AIRL under the umbrella of f-divergence. To further extend these methods, f-IRL (Ni et al., 2020) uses gradient descent to recover a stationary reward function from the expert density. In addition, recent research by the authors of (Swamy et al., 2021) suggest that various forms of IL can be understood as moment matching under different assumptions. Another line of work is based on the DICE (Nachum et al., 2019) framework. For example, ValueDICE (Kostrikov et al., 2020) utilizes the Donsker-Varadhan formulation of KL-divergence to develop an off-policy method, while SparseDICE (Camacho et al., 2021) introduces a regularizer to enable training with sparse expert data. Inspired by ValueDICE, CFIL (Freund et al., 2023) trains a pair of normalizing flows to optimize the Donsker-Varadhan representation of KL-divergence. Moreover, a variety of research efforts (K. et al., 2019; Han et al., 2022; Zeng et al., 2022; Viano et al., 2022) have been directed towards addressing specific challenges within the field. For instance, DAC (K. et al., 2019) modifies GAIL to facilitate off-policy training and concurrently tackles reward bias issues. MD-AIRL (Han et al., 2022) enhances robustness by incorporating mirror-descent into AIRL. In a further effort to improve efficiency, both ML-IRL (Zeng et al., 2022) and P^2IL (Viano et al., 2022) have been designed to relax the nested policy evaluation and cost optimization loop. Most of the above methods, while being successful in online IL, do not perform well in the single-demonstration IL setting. The reason behind this can be attributed to two primary factors. The first factor is that the majority of their objectives typically align with a min-max formulation, which could lead to unstable training, especially in situations with limited data. The second factor is inherent to their distribution-matching nature, which necessitates taking expectations over the expert distribution. Nevertheless, this process could become unreliable when dealing with sparse expert data. In contrast to these previous approaches, our methodology does not seek to match the distribution of the agent with that of the expert. This different approach avoids the issues of inaccurate expectations and unstable adversarial training.

Non-adversarial based method: Non-adversarial based methods often aim to circumvent unstable training by designating stationary rewards to guide the agent toward expert behavior. Examples include SQIL (Reddy et al., 2019), D2-Imitation (Sun et al., 2022), and ILR (Ciosek, 2022), which implement a binary reward scheme that assigns a value of 1 to expert data and 0 to agent data. These methods typically require a substantial amount of expert data to achieve optimal performance in practice. Another line of research explores a two-stage training approach, wherein a reward surrogate is first trained offline and then utilized during interaction with the environment. For instance, RED (Wang et al., 2019) estimates expert support by leveraging Random Network Distillation (Burda et al., 2019), while DRIL (Brantley et al., 2020) pretrains an ensemble of Behavior Cloning (BC) (Pomerleau, 1991) models and employs their variance as a cost function. EBIL (Liu et al., 2020) and NDI (Kim et al., 2020) employ density models, such as Energy-Based Models (EBM) (Song & Kingma, 2021) and Masked Autoencoder Density Estimation (MADE) (Germain et al., 2015), to estimate expert support density. Nevertheless, these methods necessitate a significant amount of expert data for training the offline reward surrogate, which poses challenges when applied to the single-demonstration setting. Another non-adversarial approach, PWIL (Dadashi et al., 2021), attempts to minimize discrepancy between an agent's and an expert's distributions by employing the primal form of Wasserstein distance. This method requires the computation of the Euclidean distance between every state-action pair and those of the expert, a measure that may not precisely align with the distance as defined by the Markov Decision Process (MDP). In

contrast, our method takes the properties of the underlying MDP into account. Furthermore, recent advancements such as IQ-Learn (Garg et al., 2021) and LS-IQ (Al-Hafez et al., 2023) offer a unique perspective, as they implicitly represent policy and reward using a single Q-function. Nevertheless, according to our experiments, these methods could suffer from instability during training and may not consistently perform well across various IL tasks.

A.2. Algorithm and Training Details

A.2.1. Practical Algorithm

The training process concurrently updates the transition discriminator and the SAC agent. Both the agent and expert transition data are utilized to train the SAC agent, with the agent's reward calculated using the transition discriminator. The reward calculation method involves the computation of the reward of both agent data and expert data. The agent reward is calculated by pairing the next state s_{t+1} of a transition (s_t, a_t, s_{t+1}) with "every" expert state from the demonstration, as illustrated in Fig. 3, and using the transition discriminator to calculate the reachability probability of each pair. These probabilities are then summed to yield a reward $r_t = \sum_{i=0}^{T} D_\phi(s^a_{t+1}, s^e_i)$. The expert rewards are computed in a similar manner by pairing each next state of an expert transition with every other expert state, and summing the resulting probabilities.

A.2.2. Training stabilization

To ensure stable training, a target transition discriminator, denoted as $\hat{D}$, is employed in our training process to compute the reward. $\hat{D}$ is soft-updated using the formula $\hat{D} = (1-\lambda)D + \lambda\hat{D}$, where λ is a hyperparameter set to 0.0001 in practice. The target transition discriminator helps mitigate the instability caused by SGD training, providing a more stable and consistent target for the SAC agent to learn from. This reduces overfitting and other potential sources of instability, making the training process less susceptible to fluctuations and ensuring a consistent trajectory towards convergence.

A.2.3. Algorithm detail

Algorithm 1 presents a practical training methodology of the proposed method, refering to TDIL. It takes as input the policy π of an imitator agent, an environment $\mathcal{E}$, a replay buffer B, a Transition Discriminator D, a Target Transition Discriminator $\hat{D}$, and an expert trajectory τ^e. The output is a trained optimal agent π^*. The training process is iterative, continuing until a convergence criterion is met. During each iteration, the policy π interacts with $\mathcal{E}$, and the states and actions (s_t, a_t, s_{t+1}) are stored in B. Next, D is updated based on Eq. (8) based on the stored transitions. Following this, $\hat{D}$ is soft-updated by D, which help stabilizing training. The algorithm then samples a batch of transitions from both B and the expert trajectory τ^e, and calculates the reward using $\hat{D}$. This reward is then used to update π by comparing the agent's transitions with those of the expert. Finally, π is updated using a BC loss, denoted as $L_{BC} = \text{MSE}(a \sim \pi(s^e_i), a^e_i)$, which aims to minimize the discrepancy between the agent's actions and the expert actions. Through the repetition of these steps, the TDIL algorithm trains the imitator agent π to match the expert's performance in the given environment. To satisfy the policy assumption in Section 2, the BC loss L_{BC} is included to ensure $p(\mathcal{O}_t = 1|s_t) = \max_a p(\mathcal{O}_t = 1|s_t, a)$.

Algorithm 1 TDIL: IL via Transition Discriminator

Input : Imitator Agent π, Environment $\mathcal{E}$, Replay Buffer B, Transition Discriminator D, Target Transition Discriminator $\hat{D}$, Expert Trajectory τ^e
Output : Trained optimal agent π^*

while not converge **do**
  π interacts with $\mathcal{E}$, storing s_t, a_t, s_{t+1} in B
  Update D with Eq. (8)
  Soft-update $\hat{D}$ with D
  Sample one batch of transition from B and τ^e, and calculate the reward with $\hat{D}$
  Update π using sampled agent transitions and expert transitions with calculated reward
  Update π with L_{BC}
end

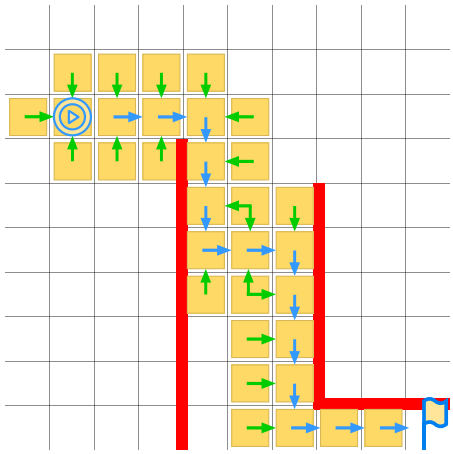


Figure A1. Visualization of the relationship between expert demonstration and expert proximity in the grid-world

A.3. Experimental Setups

A.3.1. Model Architecture of TDIL

In this section, we provide the implementation details of TDIL. The backbone of TDIL is built upon the Soft Actor-Critic (SAC) framework. The actor and critic networks in SAC are implemented as neural networks with three hidden layers and rectified linear unit (ReLU) activation functions. Each of these hidden layers consists of 256 nodes. The actor's output is first projected to $[-1, 1]$ using a hyperbolic tangent (tanh) function, and then scaled to the value range required by the environments.

A.3.2. Implementation of R_{IRL}

The implementation of R_{IRL} leverages the discriminator in GAIL. This demonstrates that, even when paired with a basic IRL reward function, the proposed R_{TDIL} can effectively guides the agent toward expert proximity and learn the expert behavior, leading to expert-level performance.

A.3.3. Code Implementation and Hardware Configuration

The code implementation and expert data used in this work are available on this GitHub repository. The computational requirements for the experiments presented in Section 5 is elaborated in Table A1.

Table A1. The hardware specification used to perform our experiments.

Hardware	Specification
RAM	128GB
CPU	AMD Ryzen Threadripper 3990X 64-Core Processor
GPU	NVIDIA GeForce RTX 3090

A.4. Additional Experiments

In this section, we provide additional experimental results and discussions. In Section A.4.1, we present the training curves of the proposed and the baseline methods to demonstrate the performance and stability of different algorithms. In Section A.4.3, we compare TDIL with baselines trained with additional BC loss. In Section A.4.4 we present the success rate curve of the proposed and the baseline method in Adroit Hand environment. In Section A.4.5, we give detailed analysis on the relative rewards for blind model selection. In Section A.4.6, we offer the evaluation results the models selected according to different blind selection metrics during training for demonstrating the effectiveness of the proposed blind selection method. In Section A.4.7, we examine the influences of the hard negative samples on the performance of the

Table A2. Performance evaluation of different methodologies using the oracle model selection.

	BC	GAIL	f-IRL	PWIL	CFIL	Ours (R_{TDIL}+ BC)	Ours (R_{agg})	Expert
HalfCheetah-v3	211 ± 49	693 ± 158	14,560 ± 823	11,460 ± 4,774	13,636 ± 1,695	**15,666** ± 85	15,624 ± 119	15,251
Hopper-v3	507 ± 161	3,209 ± 372	3,693 ± 162	3,849 ± 209	**4,131** ± 34	4,115 ± 14	4,115 ± 14	4,114
Ant-v3	990 ± 6	966 ± 22	5,597 ± 194	5,579 ± 314	5,812 ± 2,692	6,434 ± 66	**6,837** ± 19	6,561
Humanoid-v3	429 ± 20	582 ± 52	N/A	5,499 ± 94	5,354 ± 337	5,758 ± 173	**6,302** ± 136	5,855
Walker2d-v3	299 ± 75	554 ± 217	4,746 ± 316	3,391 ± 1,873	2,402 ± 949	6312 ± 47	**6334** ± 10	6,123

Table A3. TDIL when using multiple expert demonstrations

	1 Demo	2 Demo	3 Demo
HalfCheetah-v3	15,624 ± 119 (Expert: 15,251)	15,711 ± 124 (Expert: 15,200)	15,554 ± 293 (Expert: 15,197)
Hopper-v3	4,115 ± 14 (Expert: 4,114)	4,057 ± 65 (Expert: 4,194)	4,068 ± 57 (Expert: 4,188)
Ant-v3	6,837 ± 19 (Expert: 6,561)	6,486 ± 308 (Expert: 6,417)	6,341 ± 249 (Expert: 6,445)
Humanoid-v3	6,302 ± 136 (Expert: 5,855)	5,887 ± 308 (Expert: 5,926)	6,035 ± 134 (Expert: 5,920)
Walker2d-v3	6,334 ± 10 (Expert: 6,123)	6,308 ± 52 (Expert: 6,095)	6,252 ± 104 (Expert: 6,074)

transition discriminators under various scenarios. Finally in Section A.4.8 and Section A.4.9, we investigate the influence of different choices of the hyper-parameter α and β respectively.

A.4.1. Training Curves

Fig. A2 presents the training curves of TDIL as well as the other baseline methods, including BC, GAIL, f-IRL, PWIL, and CFIL. It is worth noting that the optimization of CFIL in the *HalfCheetah-v3* environment is numerically unstable as its output values sometimes become NaN during the training process. As a result, the training curve of CFIL in the *HalfCheetah-v3* environment can only be plotted partially. Fig. A2 demonstrates that TDIL is capable of reaching the expert level and exhibits a consistently stable training process across different environments compared to the other baselines.

A.4.2. Experiments on Using Multiple Expert Demonstrations

The TDIL method is designed to address the challenging limitations of the single-demo IL setting. TDIL can be regarded as an approach that leverages all available information from the expert data under the constrained condition of limited expert demonstration. However, it is not restricted to single-demo IL. Although providing more expert demonstrations might be beneficial, it does not significantly affect the performance. This is because TDIL can achieve expert-level performance with only a single expert demonstration, as shown in Table A3. Furthermore, if more expert demonstrations are available, it may not be necessary to learn a transition discriminator, and other state-of-the-art IL techniques can be employed. These techniques, however, may not adequately address the single-demo problem that TDIL is specifically designed to tackle.

A.4.3. Performance comparison between TDIL and baselines with BC loss

We have conducted additional experiments to provide a more comprehensive analysis on adding BC loss into the training process of baselines. Table A4 presents the performance of CFIL, PWIL, and TDIL with BC loss, directly compared with training the agent with R_{TDIL} and BC loss. Notably, some baselines demonstrate improved performance with BC loss, yet

Table A4. Performance of baselines with BC loss.

	CFIL w/ BC	PWIL w/ BC	f-IRL w/ BC	TDIL w/ BC
HalfCheetah-v3	14,853	4,679	13,638	15,666
Ant-v3	4,683	5,925	5,337	6,434
Humanoid-v3	5,343	5,294	N/A	5,758
Walker2d-v3	6,286	5,489	4,403	6,312

TDIL consistently outperforms all baselines. It is noteworthy that CFIL exhibited a substantial performance boost with BC loss in the Walker2d-v3 environment. However, it is crucial to acknowledge that CFIL encountered numerical issues, specifically the occurrence of actor output becoming NaN in the middle of training across all environments. This highlights potential instability in CFIL algorithm.

A.4.4. Experiments in Adroit hand environment

In Fig. A3, we present the experiment in the AdroitHandDoor environment. The AdroitHandDoor environment is a component of the Adroit manipulation platform, featuring a Shadow Dexterous Hand attached to a free arm with up to 30 actuated degrees of freedom (Rajeswaran et al., 2017). We do not evaluate TDIL on the other adroit tasks since the agent is required to achieve different goals encapsulated within the state feature. In such environments, agents cannot learn the meaning of different goals if only one expert demonstration is offered. This limitation arises because all expert states in the provided single demonstration inherently possess the same goal, which restricts the agent's comprehension of the goal feature. As a result, the agent might not learn the goal feature adequately, and this can result in a policy that fails to condition on the goal effectively. For instance, the agent could mimic an expert trajectory without adapting to changes in the goal.

In the AdroitHandDoor-v1 scenario, the task involves undoing a latch and swinging open a door with a biased torque that keeps it closed. The environment, based on a 28-degree-of-freedom system, includes a 24-degree-of-freedom ShadowHand and a 4-degree-of-freedom arm. The action space is represented as a Box(-1.0, 1.0, (28,), float32), with control actions specifying absolute angular positions of the hand joints. The observation space is a Box(-inf, inf, (39,), float64), containing information on finger joint angles, palm pose, and the state of the latch and door, Fig. 7 illustrates the task.

The episode's time step limit is set at 200. During the testing phase, the agent undergoes perturbation through five time-steps of random actions in the beginning of the episode to enhance difficulty and introduce stochasticity. In comparison to BC and two of the top-performing baselines from the main experiment, the results demonstrate that TDIL attains an expert-level performance within 1 million steps, surpassing the performance of BC, PWIL, and CFIL.

A.4.5. Exploring Relative Rewards for Blind Model Selection

Blind model selection refers to the process of choosing the optimal model checkpoint throughout the training phase, holds significant importance in the field of IL. In IL, it is generally assumed that obtaining the ground truth reward from the environment is unfeasible, even during testing. This issue, often neglected in prior research, warrants considerable attention. Although the reward signals proposed in this work, denoted as R_{TDIL}, can effectively train the agent, they may not be ideally suited for blind model selection. As training progresses, a potential decrease in the agent's raw rewards is observed. The reduction in raw agent reward may not necessarily signify a decrease in agent's performance; rather, it mirrors the enhanced accuracy of the transition discriminator. As a result, it becomes imperative to establish an indicator that is strongly correlated with the ground truth reward. Such an indicator would facilitate reliable model selection in IL. To meet this requirement, we introduce the concept of 'relative reward,' which is denoted as r_{relative} and is defined as follows:

$$r_{\text{relative}} = r_{\text{raw agent}} / r_{\text{raw expert}}, \tag{A1}$$

where $r_{\text{raw agent}} = \sum_{t=0}^{\tilde{T}} R_{\text{TDIL}}(s_t, a_t)$ and $r_{\text{raw expert}} = \sum_{t=0}^{T} R_{\text{TDIL}}(s_t^e, a_t^e)$ are the total rewards along the agent's and expert's trajectories, and $\tilde{T}$ is the length of the agent's trajectory. As the transition discriminator may improve its accuracy during training, our aim is to mitigate the influence of its accuracy on reflecting the true extent of reward signals. In an ideal scenario, the reward for expert actions should be higher, while those outside the expert support should be lower. With this in mind, the essence of Eq. (A1) is to calculate the relative reward by dividing the raw agent reward, derived from the transition discriminator, by the raw expert reward, also derived from the transition discriminator. This process aids in neutralizing the impact of potential inaccuracies of the transition discriminator. The rationale behind this approach is the presumption that the inaccuracies in the transition discriminator would affect both the raw agent reward and the raw expert reward in a similar fashion. Hence, when the raw agent reward is divided by the raw expert reward, any inaccuracies that potentially exist in the transition discriminator should theoretically cancel out. This is because these inaccuracies are likely to proportionally affect the numerator (i.e., the raw agent reward) and the denominator (i.e., the raw expert reward) of the division. For example, if the transition discriminator is consistently underestimating or overestimating the rewards, both the raw agent reward and the raw expert reward would be underestimated or overestimated to a comparable extent. As a result, their ratio (i.e., the relative reward) should still provide a reliable comparison of agent performance relative to the expert, even if the absolute reward values are incorrect. This approach, therefore, helps to render the reward calculation more robust to the inaccuracies of the transition discriminator, and enhances the reliability of the model selection process in the single-demonstration IL context.

Table A5. Performance decrease ratios of different methods in the blind model selection scenario.

	BC (Pomerleau, 1991)	f-IRL (Ni et al., 2020)	PWIL (Dadashi et al., 2021)	CFIL (Freund et al., 2023)	Ours
HalfCheetah-v3	-0.75 $\pm$ 0.34	-0.27 $\pm$ 0.27	-0.17 $\pm$ 0.15	-0.07 $\pm$ 0.00	**-0.02** $\pm$ 0.01
Hopper-v3	-0.27 $\pm$ 0.23	-0.32 $\pm$ 0.26	-0.31 $\pm$ 0.29	**-0.04** $\pm$ 0.02	**-0.04** $\pm$ 0.05
Ant-v3	-0.73 $\pm$ 0.03	-0.18 $\pm$ 0.16	-0.13 $\pm$ 0.18	**-0.03** $\pm$ 0.01	**-0.03** $\pm$ 0.01
Humanoid-v3	-0.18 $\pm$ 0.16	N/A	-0.18 $\pm$ 0.16	**-0.03** $\pm$ 0.03	-0.04 $\pm$ 0.03
Walker2d-v3	-0.45 $\pm$ 0.07	-0.92 $\pm$ 0.08	-0.45 $\pm$ 0.26	-0.50 $\pm$ 0.31	**-0.03** $\pm$ 0.04

A.4.6. Blind Model Selection Experiments

To further substantiate the efficacy of utilizing relative rewards in blind model selection, we performed a MuJoCo experiment in which the optimal testing model was selected without any access to the environmental ground truth rewards. In this experiment, our method used relative rewards as an indicator. In contrast, PWIL employed the Wasserstein distance, following the methodology of the original paper. For the remaining methods, which did not provide an indicator for model selection in their original manuscripts, we chose the model with the lowest policy loss. Table A5 presents the ratio of performance decrease of each method, which is calculated according to $\frac{blind\ result\ -\ oracle\ result}{oracle\ result}$. The results reveal that our proposed method outperforms both policy loss-based model selection and Wasserstein distance-based model selection schemes. This outcome suggests that relative rewards can effectively guide the selection of the best model, and provides a valuable insight that can be applied in future single-demonstration IL research to develop similar indicators for practical use.

To demonstrate the effectiveness of the blind model selection strategy over the model selection methods adopted by the baselines, we compare the returns obtained using the proposed strategy and the baseline methods along with the highest testing return achieved by each agent during its training process. Fig. A4 presents the results of the above setting. In the figure, the blue and red curves represent the total return obtained by each agent and the model selection strategy metric employed by each baseline, respectively. In addition, the solid and the dashed lines depict the highest testing return achieved by each agent during its training process and the return determined by the blind selection strategy, respectively. It is observed that our method is effective in selecting a model with high performance, as the distance between the solid and the dashed lines shown in Fig. A4 (d) is the closest as compared to those depicted in Figs. A4 (a), (b), and (c). Please note that the returns of the baseline methods can be derived using either the returns of the agent in the last step or the returns selected according to their respective blind selection metrics. In Table A5 of the main manuscript, we report the higher returns achieved by the baseline methods.

A.4.7. An Analysis of the Accuracy of the Transition Discriminator

Fig. A5 illustrates the accuracy of the transition discriminator evaluated on positive samples, easy negative samples, and hard negative samples. Of particular interest is the accuracy of the hard negative samples. In *HalfCheetah-v3* and *Ant-v3* (i.e., Figs. A5(a) and (b), respectively), the transition discriminator trained without the use of hard negative samples demonstrates similar accuracy to the one trained with hard negative samples. However, in *Humanoid-v3* (Fig. A5(c)), the transition discriminator trained without hard negative samples exhibits significantly lower accuracy compared to the one trained with hard negative samples. These findings substantiate the assumption presented in Section 5.4, which suggests that the set of hard negative samples falls within the subset of easy negative samples. In relatively less complex environments, the agent can extract the information embodied in hard negative samples even when training exclusively with easy negative samples. However, this scenario is less probable in the more demanding *Humanoid-v3* environment, leading to the observed discrepancy in accuracy between the two training settings. These experimental results highlight the importance of incorporating hard negative samples, particularly in complex environments, to improve the accuracy and effectiveness of the proposed transition discriminator.

A.4.8. Sensitivity analysis on the hyper-parameter α

We have addressed the sensitivity of the proposed TDIL algorithm to different values of the hyper-parameter α by presenting the corresponding training curves in Fig. A6. The introduction of the balancing factor α for positive and negative samples aims to mitigate the impact of false negative samples within the pool of easy negative samples. These easy negative samples are composed of pairs of individually randomly sampled states from the replay buffer, and there exists a chance that these

Table A6. Accuracy of the transition discriminator when trained with different α. "p" stands for positive data's accuracy; "en" stands for easy negative data's accuracy; "hn" stands for hard negative data's accuracy;

	0.5(p)	0.5(en)	0.5(hn)	0.67(p)	0.67(en)	0.67(hn)	0.9(p)	0.9(en)	0.9(hn)	0.99(p)	0.99(en)	0.99(hn)
HalfCheetah-v3	1.0	0.997	0.998	1.0	0.996	1.0	1.0	0.997	0.996	1.0	0.992	0.996
Hopper-v3	0.996	0.996	0.992	0.996	0.996	0.996	0.996	0.996	0.992	1.0	0.996	0.996
Ant-v3	0.996	1.0	1.0	1.0	1.0	1.0	1.0	0.996	1.0	1.0	0.992	0.992
Humanoid-v3	1.0	1.0	1.0	1.0	0.996	0.996	0.996	0.996	0.996	1.0	0.99	0.988
Walker2d-v3	0.996	0.996	0.996	0.992	0.996	0.996	0.996	0.992	0.99	1.0	0.992	0.988

Table A7. Performance of TDIL under different β value selection.

	0+BC	0	0.1	0.2	0.5	0.8	0.9	0.95	0.99	1.0	Expert
HalfCheetah-v3	**15,666**	12,630	15,100	15,541	15,479	15,612	15,624	15,462	15,529	9,791	15,251
Hopper-v3	4,115	3,890	4,124	4,126	**4,162**	4,128	4,115	1,887	3,232	1,950	4,114
Ant-v3	6,434	3,995	6,358	6,513	6,467	6,611	**6,837**	6,560	6,506	4,216	6,561
Humanoid-v3	5,758	5,575	6,288	**6,352**	6,312	6,325	6,302	5,703	5,235	1,826	5,855
Walker2d-v3	6,312	6,281	6,251	6,204	6,266	**6,346**	6,334	6,296	6,098	1,769	6,123

pairs may form valid transitions under the Markov Decision Process (MDP), effectively becoming positive samples.

To safeguard the training of the transition discriminator against the adverse effects of false negatives, we assign a smaller weight to negative samples compared to positive samples. Experimental results demonstrate that when α is set to small values (e.g., 0.5, 0.67), the agent takes longer to reach optimal performance in certain environments. Conversely, when the value of α is set to 0.99, the algorithm consistently performs well across various environments. This observation underscores the importance of choosing the hyper-parameter α to ensure optimal and robust performance of the TDIL algorithm.

A.4.9. Experimental result on different choices of hyper-parameter β

The ablation study on the hyper-parameter β is comprehensively presented in Table A7, shedding light on its impact within the overall reward function. By aggregating R_{TDIL} and R_{IRL} with a judicious selection of β, the agent consistently attains expert-level performance guided by this composite reward. Experimental findings suggest that values of β within the range of [0.1, 0.9] yield favorable results across various MuJoCo environments. This observation underscores the intrinsic ability and efficacy of the reward function R_{agg}.

Importantly, these results indicate that setting β to zero, as done in the main experiments, can still produce effective outcomes, particularly when approximating the effect of R_{IRL} through BC loss. This pragmatic approach not only maintains computational efficiency but also highlights the adaptability and robustness of the proposed TDIL method, even when certain components, such as β, are tuned or simplified for specific experimental contexts.

A.5. Multi-Step Expert Proximity

In the main manuscript, the expert reachability indicator $\tilde{\mathcal{O}}_t$ is only defined to consider the transition to expert states within a single timestep. We could generalize the reachability indicator to multiple timesteps by defining $\tilde{\mathcal{O}}_t^{(k)}$, where it determines whether the state s_t can reach an expert state by selecting a series of k actions. Formally, we define the following:

$$p(\tilde{\mathcal{O}}_t^{(k)}|s_t,a_t) \stackrel{\text{def}}{=} \begin{cases} \int_{\mathcal{S}} p(s_{t+1}|s_t,a_t)p(\mathcal{O}_{t+1}|s_{t+1})ds_{t+1} & \text{if } k=1, \\ \int_{\mathcal{S}} p(s_{t+1}|s_t,a_t)p(\tilde{\mathcal{O}}_{t+1}^{(k-1)}|s_{t+1})ds_{t+1} & \text{if } k\in\{2,\dots,T\}, \end{cases} \tag{A2}$$

The value of $p(\tilde{\mathcal{O}}_t^{(k)}|s_t)$ can be calculated as in the main manuscript. The surrogate reward functions corresponding to the indicators are defined as follows:

$$R_{\text{TDIL}}^{(k)}(s_t, a_t) \stackrel{\text{def}}{=} \mathbb{E}_{s_{t+1}\sim p(s_{t+1}|s_t,a_t)}\Big[p(\tilde{\mathcal{O}}_{t+1}^{(k)}|s_{t+1})\Big]. \tag{A3}$$

Each surrogate reward functions can be approximated by $D_{\phi^*}^{(k)}(s_i, s_j)$ defined as:

$$D_{\phi^*}^{(k)}(s_i, s_j) \stackrel{\text{def}}{=} \max_{a_i,\ldots,a_{i+k-1}} \mathbb{1}\left[\prod_{j=i}^{i+k-1} P(s_j|s_i, a_j) > 0\right]. \tag{A4}$$

The total reward function R_{agg} can then be re-defined based on the weighted sum of the surroagte reward functions $R_{\text{TDIL}}^{(k)}$ across all k.

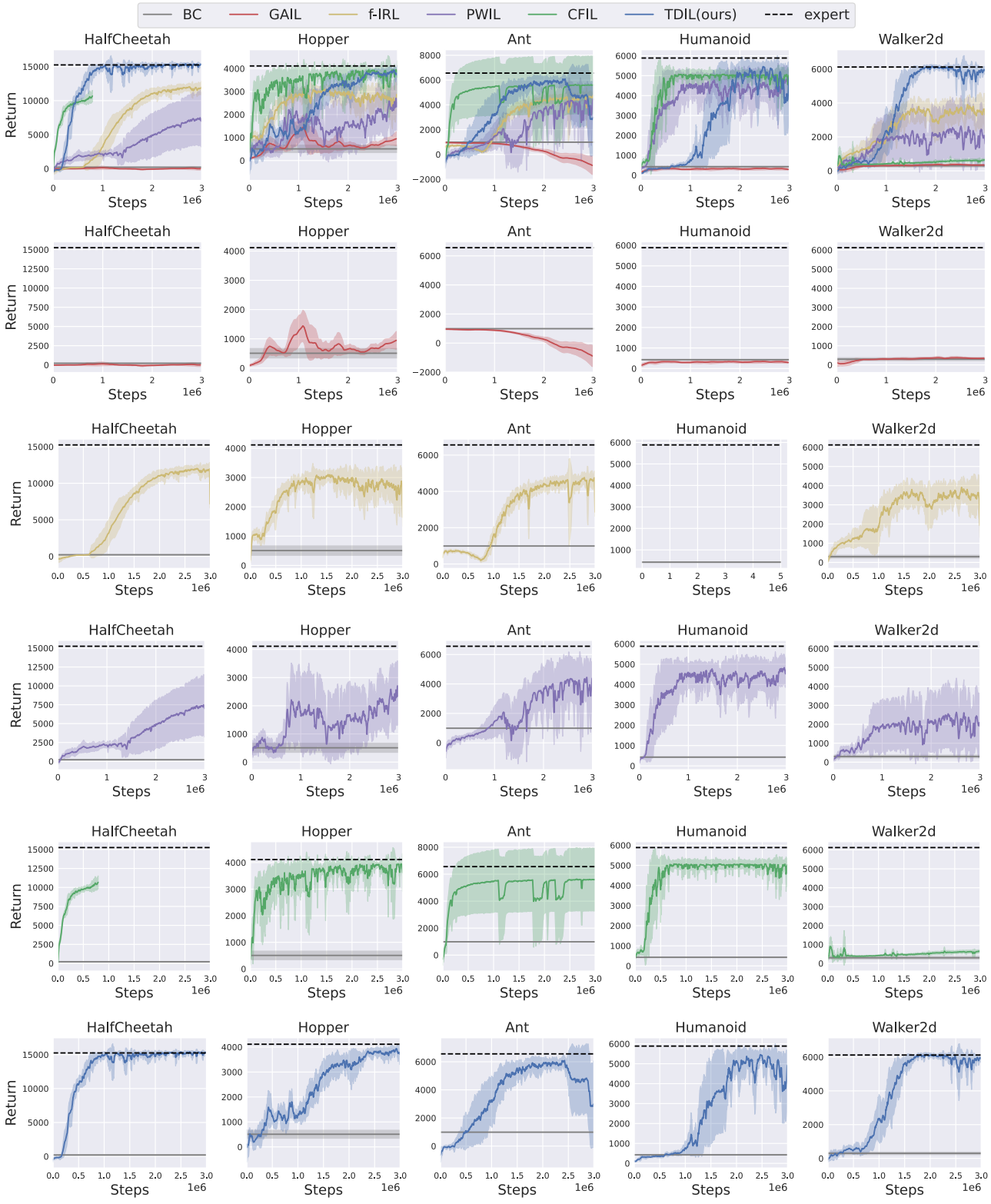


Figure A2. The training curves of BC, GAIL, f-IRL, PWIL, CFIL, and TDIL. These curves represent the means and the standard deviations of five independent runs conducted with different random seeds

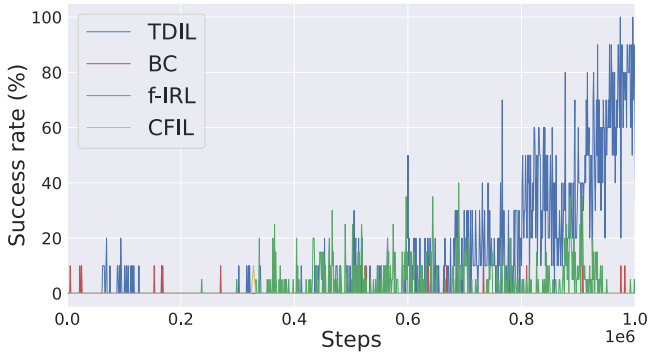
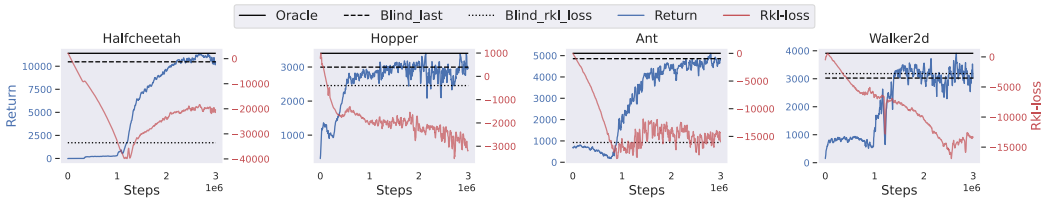
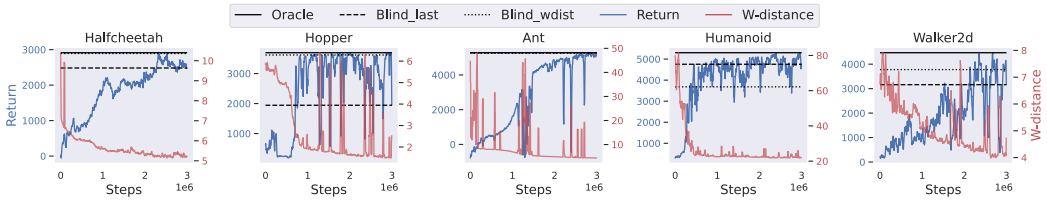


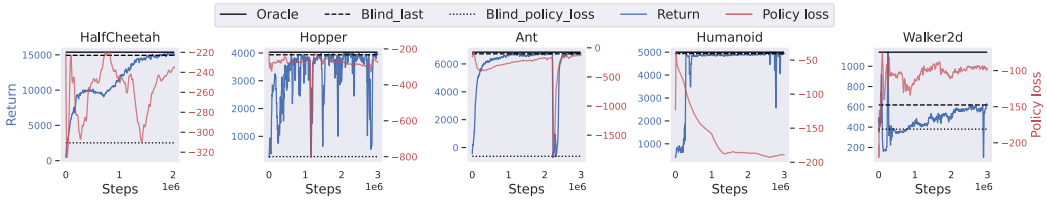
Figure A3. Comparing the success rates of TDIL, BC, f-IRL and CFIL in the AdroitHandDoor-v1 environment.



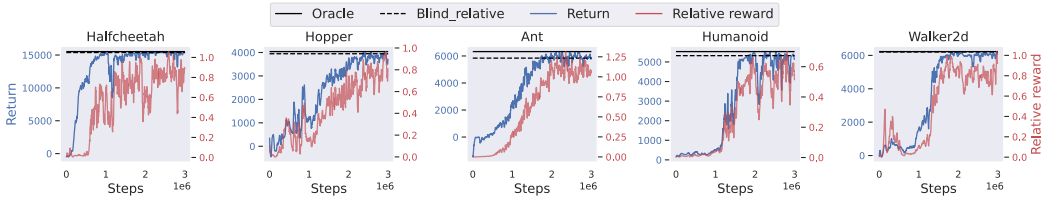
(a) f-IRL



(b) PWIL

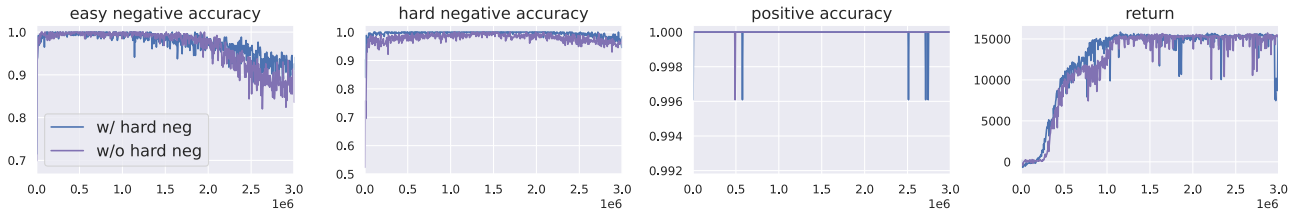


(c) CFIL

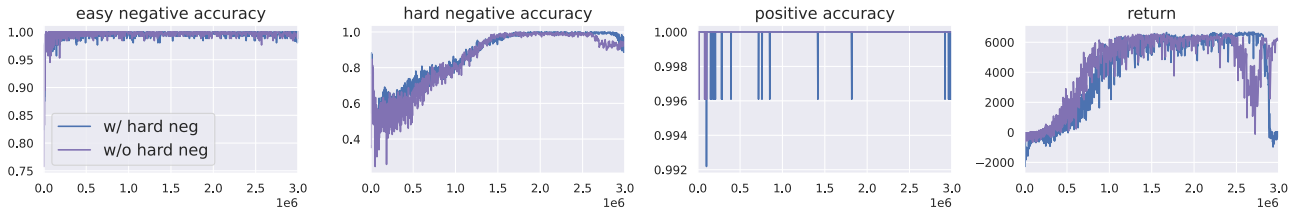


(d) TDIL (ours)

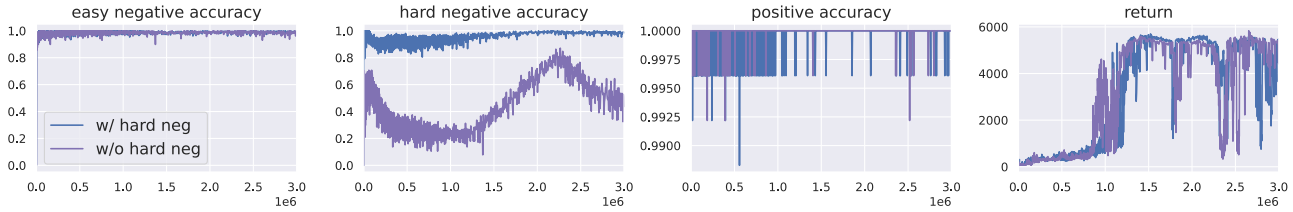
Figure A4. The training curves and the blind selection results of f-IRL, PWIL, CFIL, and TDIL (ours). The oracle line represents the highest evaluation return achieved during training. The Blind_last line depicts the evaluation return achieved by the agent at the end of the training phase. The Blind_{rkl_loss, wdist, policy_loss, relative} lines correspond to the evaluation returns determined based on the reverse KL loss, W-distance, policy-loss, and our proposed relative reward, respectively.



(a) HalfCheetah



(b) Ant



(c) Humanoid

Figure A5. An analysis of the accuracy of the transition discriminator in different environments

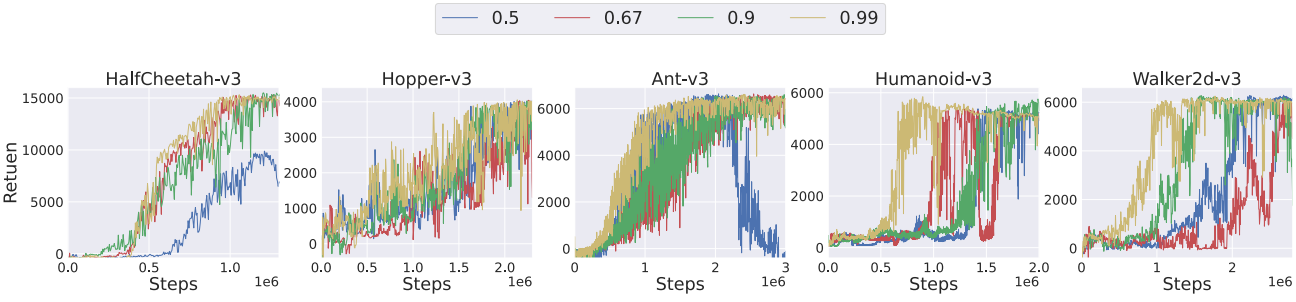


Figure A6. Performance of different α selection