DiffAIL: Diffusion Adversarial Imitation Learning

Bingzheng Wang, Guoqiang Wu*, Teng Pang, Yan Zhang, Yilong Yin*

Shandong University
binzhwang@gmail.com, guoqiangwu@sdu.edu.cn ,
silencept7@gmail.com, yannzhang9@gmail.com, ylyin@sdu.edu.cn

Abstract

Imitation learning aims to solve the problem of defining reward functions in real-world decision-making tasks. The current popular approach is the Adversarial Imitation Learning (AIL) framework, which matches expert state-action occupancy measures to obtain a surrogate reward for forward reinforcement learning. However, the traditional discriminator is a simple binary classifier and doesn't learn an accurate distribution, which may result in failing to identify expert-level state-action pairs induced by the policy interacting with the environment. To address this issue, we propose a method named diffusion adversarial imitation learning (DiffAIL), which introduces the diffusion model into the AIL framework. Specifically, DiffAIL models the state-action pairs as unconditional diffusion models and uses diffusion loss as part of the discriminator's learning objective, which enables the discriminator to capture better expert demonstrations and improve generalization. Experimentally, the results show that our method achieves state-of-the-art performance and significantly surpasses expert demonstration on two benchmark tasks, including the standard state-action setting and state-only settings.

Introduction

Deep Reinforcement Learning has achieved significant success in many decision-making tasks, including AlphaGo (Silver et al. 2016), Atari games (Mnih et al. 2013), MuJoCo environment tasks (Todorov, Erez, and Tassa 2012), and robot control tasks (Mnih et al. 2015), where these tasks are typically defined with clear reward functions to guide the agent for decision-making. However, in real-world scenarios, the reward function is regularly challenging to obtain or define, such as in the case of autonomous driving, where it is complicated to delineate what behavior is beneficial for the agent. The emergence of *Imitation Learning* provides a practical solution to the problem of inaccessible rewards. In imitation learning, the agent does not rely on actual rewards but instead utilizes expert demonstration data to learn a similar expert policy.

A relatively old method in imitation learning is clone learning (Pomerleau 1991), which uses supervised learning to learn from expert data. Although such methods are straightforward to implement, they are prone to serious extrapolation errors when visiting out-of-distribution data while practically interacting with environments. To alleviate the aforementioned errors, Dagger (Ross, Gordon, and Bagnell 2011) proposes access to expert policies, where the agent continuously interacts with the environment online and asks for expert-level actions from the expert policy to expand the dataset. AdapMen (Liu et al. 2023) proposes an active imitation learning framework based on teacher-student interaction, and theoretical analysis shows that it can avoid compounding errors under mild conditions.

To effectively tackle the aforementioned extrapolation error, Adversarial imitation learning (AIL) (Ho and Ermon 2016; Fu, Luo, and Levine 2017; Kostrikov et al. 2018; Ghasemipour, Zemel, and Gu 2020; Zhang et al. 2020; Garg et al. 2021; Zhang et al. 2022) has become the most popular approach in imitation learning. Rather than minimizing the divergence between the expert policy and the agent policy, AIL employs online interactive learning to focus on minimizing the divergence between the joint state-action distribution induced by the expert policy and the learned policy. However, typical AIL methods often use a simplistic discriminator that does not learn a distribution, which may not accurately classify specific expert-level actions generated by the policy during interactions with the environment. This can result in the naive discriminator failing to distinguish expert-level behaviors generated by the policy, potentially hindering agent learning. Therefore, a powerful discriminator that can accurately capture the distribution of expert data is crucial for AIL.

Currently, diffusion model (Ho, Jain, and Abbeel 2020) processes a powerful distribution matching capability and has achieved remarkable success in image generation, surpassing other popular generative models such as GAN (Goodfellow et al. 2014) and VAE (Kingma and Welling 2013) in producing higher-quality and more diverse samples.

Based on diffusion model's powerful ability to capture data distribution, we propose a new method called Diffusion Adversarial Imitation Learning (DiffAIL) in this work. DiffAIL employs the same framework as traditional AIL but incorporates the diffusion model to model joint state-action distribution. Therefore, DiffAIL enables the discriminator to accurately capture expert demonstration and im-

*Corresponding authors.

prove generalization, facilitating the successful identification of expert-level behaviors generated by the policy during interactions with the environment. Furthermore, we conducted experiments on representatively continuous control tasks in Mujoco, including the standard state-action setting and state-only setting. Surprisingly in both settings, we share the hyperparameters instead of readjusting them. The experiment results show that our method can achieve state-of-the-art (SOTA) performance and significantly surpass expert demonstration on these two benchmark tasks.

Overall, our contributions can be summarized as follows:

- We propose the DiffAIL method, which combines the diffusion model into AIL to improve the discriminator ability of distribution capturing.
- The experimental results show that DiffAIL can achieve SOTA performance, including standard state-action and state-only settings.

Preliminaries

Problem Setting

A sequential decision-making problem in reinforcement learning can be modeled as a Markov Decision Process (MDP) defined as a tuple $M = (S, A, \rho_0, P, r, \gamma)$, where S denotes the state space, A denotes the action space, ρ_0 denotes the initial state distribution, $P : S \times A \times S \to [0, 1]$ describes the environment's dynamic model by specifying the state transition function, $r : S \times A \to \mathbb{R}$ denotes the reward function, and $\gamma \in [0, 1]$ denotes the discount factor used to weigh the importance of future rewards. We represent the agent policy as $\pi : S \to A$. Policy $\pi(a|s)$ interacts with the environment to generate transitions (s_t, a_t, r_t, s_{t+1}),where t denotes the timestep, $s_0 \sim \rho_0, a_t \sim \pi(\cdot|s_t), r_t \sim r(s_t, a_t), s_{t+1} \sim p(\cdot|s_t, a_t)$. The goal of reinforcement learning is to learn a policy $\pi(a|s)$ maximizing cumulative discount rewards.

In imitation learning, the environment reward signal is not accessible. Rather, we can access expert demonstrations $D = \{(s_t, a_t)\}_{t=1}^{k} \sim \pi_e$ given by the unknown expert policy π_e. Imitation learning aims to learn a policy π that can recover π_e based on expert demonstrations without relying on reward signals. This setting can be purely offline to learn with only expert demonstrations or additionally online to interact with the environment (reward unavailable) by behavior policy.

Adversarial Imitation Learning

For AIL, it aims to find an optimal policy that minimizes the state-action distribution divergence induced by the agent policy and the expert policy, as measured by the occupancy measure:

$$\pi^* = \arg\min_{\pi} D_f(d_\pi(s, a)||d_{\pi_e}(s, a)), \tag{1}$$

where $d_\pi(s, a) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t p(s_t = s, a_t = a)$ denotes state-action distribution of π_θ, $(1 - \gamma)$ term denotes normalization factor. Similarly, $d_{\pi_e}(s, a)$ denotes expert demonstration distribution with a similar form. D_f can be an arbitrary distance formulation with the corresponding min-max target through the corresponding dual representation, which is also applied in f-gan (Nowozin, Cseke, and Tomioka 2016). Gail (Ho and Ermon 2016) adopts the Jensen-Shannon (JS) divergence as the chosen distance metric D_f. By leveraging the dual representation of JS divergence, we can obtain the minimax optimization objective of Gail:

$$\min_{\pi_\theta} \max_{D_\phi} \mathbb{E}_{(s,a)\sim\pi_e}[\log(D_\phi(s, a))] + \mathbb{E}_{(s,a)\sim\pi_\theta}[\log(1 - D_\phi(s, a))]. \tag{2}$$

The discriminator D_ϕ is used to identify between samples produced by the learning policy and expert demonstrations. Typically, the discriminator is a binary classifier whose output tends to be 1 for expert data and 0 for data generated by the policy. The optimal discriminator satisfies the following:

$$\log D^*(s, a) - \log(1 - D^*(s, a)) = \log \frac{d_{\pi_e}(s, a)}{d_{\pi_\theta}(s, a)}, \tag{3}$$

where $D^* = \frac{d_{\pi_e}(s,a)}{d_{\pi_e}(s,a)+d_{\pi_\theta}(s,a)}$. The log-density ratio can be directly used as a surrogate reward function for forward reinforcement learning in CFIL (Freund, Sarafian, and Kraus 2023). Other popularly used surrogate reward function settings include $R(s, a) = \log D_\phi(s, a)$ or $R(s, a) = -\log(1 - D_\phi(s, a))$ used by Gail (Ho and Ermon 2016) to minimize the Jensen-Shannon (JS) divergence and $R(s, a) = \log D_\phi(s, a) - \log(1 - D_\phi(s, a))$ used by AIRL (Fu, Luo, and Levine 2017) to minimize the reverse Kullback-Leibler (KL) divergence. Although these surrogate reward functions have some prior bias in the absorbed state (Kostrikov et al. 2018), they still achieve good results in practical use.

All of these surrogate reward functions contain the discriminator term, which means that an advanced discriminator can provide better guidance for policy learning.

With these surrogate reward functions, Gail and other imitation methods can be combined with any forward reinforcement learning algorithm for policy optimization, such as PPO (Schulman et al. 2017) or TRPO (Schulman et al. 2015) for on-policy algorithms and TD3 (Fujimoto, Hoof, and Meger 2018) or SAC (Haarnoja et al. 2018) for off-policy algorithms.

Diffusion Model

Diffusion model (Ho, Jain, and Abbeel 2020) is a latent variable model mapped to potential space using a Markov chain. In the forward diffusion process, noise is gradually added to the data x_t at each time step t using a pre-defined variance schedule β_t. As t increases, x_t gets closer to pure noise, and when $T \to \infty$, x_T becomes the standard Gaussian noise.

$$q(x_{1:T}|x_0) := \prod_{t=1}^{T} q(x_t|x_{t-1}), \tag{4}$$

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1 - \beta_t}\, x_{t-1}, \beta_t I). \tag{5}$$

By defining $\alpha_t = 1 - \beta_t$ and $\overline{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$, we can use the reparameterization trick to directly sample x_t at any time step t from x_0:

$$x_t = \sqrt{\overline{\alpha}_t}\, x_0 + \sqrt{1-\overline{\alpha}_t}\, \epsilon, \tag{6}$$

where $\epsilon \sim \mathcal{N}(0,1)$. The reverse diffusion process involves gradual sampling from the pure Gaussian noise to recover x_0. This process is modeled by latent variable models of the form $p_\phi(x_0) := \int p_\phi(x_{0:T}) dx_{1:T}$.

$$p_\phi(x_{0:T}) := p(x_T) \prod_{t=1}^{T} p_\phi(x_{t-1}|x_t), \tag{7}$$

$$p_\phi(x_{t-1}|x_t) := \mathcal{N}(x_{t-1};\ \mu_\phi(x_t,\ t), \Sigma_\phi(x_t,\ t)), \tag{8}$$

where $\mu_\phi(x_t, t) = \frac{1}{\sqrt{1-\beta_t}}(x_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\epsilon_\phi(x_t,\ t))$ and ϵ_ϕ is used to predict ϵ from x_t in the reverse process by parameter ϕ. The covariance matrix $\Sigma_\phi(x_t, t)$ can be fixed for non-trainable or parameterized for learning.

The diffusion model maximizes the log-likelihood of the predicted distribution of the model, which is given by $\mathbb{L} = \mathbb{E}_{q(x_0)}[\log p_\phi(x_0)]$. Subsequently, similar to VAE (Kingma and Welling 2013), the diffusion model maximizes the variational lower bound (VLB) defined as $\mathbb{E}_{q(x)}[\ln \frac{p_\phi(x_{0:T})}{q(x_{1:T}|x_0)}]$. The continuous derivation of VLB results in a simple loss function for the diffusion model, which can be expressed as follows:

$$L_{simple}(\phi) = \mathbb{E}_{\ t\sim\mu(1,T),\ \epsilon\sim\mathcal{N}(0,I),\ x_0\sim q(x_0)}[||\epsilon - \epsilon_\phi(\sqrt{\overline{\alpha}_t}\, x_0 + \sqrt{1-\overline{\alpha}_t}\, \epsilon,\ t)||^2], \tag{9}$$

where $\overline{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$, μ is a uniform distribution from between 1 and T and x_0 sampled from real data.

Method

In this section, we present our proposed method - DiffAIL. First, we describe how we model the unconditional diffusion process on the joint distribution of state-action pairs. Second, we incorporate the diffusion model into the discriminator of an AIL framework to improve the ability that the discriminator to capture data distribution.

Diffusion Over State-Action Pairs

To simplify the notation, we abuse notation and define $x_t = (s_i, a_i)_t$, where t denotes the time step in the diffusion process and i denotes the time at which a particular state is visited or an action in the trajectory. x_t represents the state-action pairs available in the data. Since the forward diffusion process is parameter-free, we denote our forward diffusion process as follows:

$$q(x_t|x_{t-1}) = \mathcal{N}(x_t;\ \sqrt{1-\beta_t}\, x_{t-1}, \beta_t I), \tag{10}$$

where β_t denotes the variance schedule at different time steps t. We then define the reverse process of the parameterized diffusion model as follows:

$$p_\phi(x_{0:T}|x_t) = \mathcal{N}(x_{t-1};\ \mu_\phi(x_t,\ t), \Sigma_\phi(x_t,\ t)). \tag{11}$$

In our work, the covariance matrix is fixed as $\Sigma_\phi(x_t,\ t) = \sigma_t^2 I = \beta_t I$ to be non-trainable and represented using a pre-defined time schedule β_t, following the same as in DDPM (Ho, Jain, and Abbeel 2020). According to the forward process, x_t can be derived from x_0 at any time step t; the mean can be expressed as a noisy contained function:

$$\mu_\phi(x_t,\ t) = \frac{1}{\sqrt{1-\beta_t}}(x_t - \frac{\beta_t}{\sqrt{1-\overline{\alpha}_t}}\, \epsilon_\phi(x_t,\ t)). \tag{12}$$

Diffuser over state-action pairs is ultimately modelled as predicting the noise at each time step in the reverse diffusion process:

$$L(\phi) = \mathbb{E}_{x_0\sim D, \epsilon\sim\mathcal{N}(0,I), t\sim\mu(1,T)}[Diff_\phi(x_0, \epsilon, t)], \tag{13}$$

$$Diff_\phi(x_0, \epsilon, t) = ||\epsilon - \epsilon_\phi(\sqrt{\overline{\alpha}_t} x_0 + \sqrt{1-\overline{\alpha}_t}\, \epsilon,\ t)||^2, \tag{14}$$

where μ is the uniform distribution, ϵ follows the standard Gaussian noise distribution $\mathcal{N}(0, I)$ and $\alpha_t = \prod_{s=1}^{T}(1-\beta_s)$. In this work, we do not employ the diffusion model to infer for generating samples but use its loss as a distribution matching technique to improve the discriminator's ability to identify the expert demonstrations and policy data. Therefore, practically, $Diff_\phi$ can refer to either expert demonstrations $(s_i, a_i) \sim d_{\pi_e}(s, a)$ or data generated from interactions with the environment by policy $(s_i, a_i) \sim d_\pi(s, a)$.

Although many studies (Song, Meng, and Ermon 2020; Lu et al. 2022a,b; Zheng et al. 2023) have focused on designing accelerated sampling algorithms for diffusion models to reduce the time cost of generating samples while maintaining high sample quality, we found that only a few tens of diffusion steps were sufficient to achieve excellent performance when modeling various environment tasks in MuJoCo, unlike image generation tasks that require thousands of sampling steps. Therefore, in our work, the primitive form of the diffusion model (Ho, Jain, and Abbeel 2020) is adequate and efficient.

Diffusion Adversarial Imitation Learning

Next, we describe how diffusion loss is integrated into the AIL framework to improve discriminator generalization.

We introduce the diffusion model into adversarial imitation learning, which combines the simple loss function of the diffusion model with the traditional discriminator structure of AIL. For convenience, we abbreviate x_0 as x in the following text. The objective function used in our method is the same as Gail (Ho and Ermon 2016):

$$\min_{\pi_\theta}\max_{D_\phi} \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I), t\sim\mu(1,T)}[\mathbb{E}_{x\sim\pi_e}[\log(D_\phi(x,\epsilon,t))] + \mathbb{E}_{x\sim\pi_\theta}[\log(1 - D_\phi(x,\epsilon,t))]], \tag{15}$$

where the $D_\phi(x, \epsilon, t)$ is a discrimination composed of diffusion noise loss by parameter ϕ, and the policy network's parameters are represented as θ. At the inner level maximization discriminator provides higher values for expert data demonstrations and penalizes all other data areas. The specific formulation of $D_\phi(x, \epsilon, t)$ is defined as follows:

$$\begin{aligned} D_\phi(x, \epsilon, t) &= \exp\left(-Diff_\phi(x, \epsilon, t)\right) \\ &= \exp\left(-||\epsilon - \epsilon_\phi(\sqrt{\overline{\alpha}_i}\, x + \sqrt{1-\overline{\alpha}_t}\, \epsilon,\ t)||^2\right). \end{aligned} \tag{16}$$

Task	Hopper	HalfCheetah	Ant	Walker2d
Expert	3402	4463	4228	6717
BC	2376.22 ± 754.80	2781.85 ± 1143.11	2942.24 ± 344.62	1091.46 ± 187.60
Gail	2915.67 ± 328.12	4383.44 ± 62.08	4530.43 ± 133.30	1942.81 ± 957.68
Valuedice	2455.25 ± 658.43	4734.59 ± 229.67	3781.56 ± 571.15	4606.50 ± 927.80
CFIL	3234.61 ± 266.62	4976.73 ± 60.41	4222.07 ± 201.32	5039.77 ± 307.17
DiffAIL	**3382.03 ± 142.86**	**5362.25 ± 96.92**	**5142.60 ± 90.05**	**6292.28 ± 97.65**

Table 1: Learned policy best performance during training for different sota imitation learning algorithms with 1 trajectory over 5 seeds in the standard state-action setting.

We apply the exponential operation on the negative diffusion loss, which allows DiffAIL to restrict the discriminator output strictly within $(0, 1)$ and continue satisfying the optimization objective of minimizing the JS divergence. This is consistent with the logit value range after the sigmoid function in the original Gail and coincides with the monotonicity requirement.

By combining Eq. (15) and Eq. (16), we can obtain our final optimization objective, which utilizes the diffusion loss as the part of discriminator in AIL:

$$\min_{\pi_\theta}\max_{D_\phi}\mathbb{E}_{\epsilon\sim\mathcal{N},t\sim\mu}[\mathbb{E}_{x\sim\pi_e}[\log(\exp(-Diff_\phi(x,\epsilon,t)))] + \mathbb{E}_{x\sim\pi_\theta}[\log(1-\exp(-Diff_\phi(x,\epsilon,t)))]]. \quad (17)$$

In this adversarial framework, training the diffusion model loss is an adversarial process. $Diff_\phi(x,\epsilon,t)$ will minimize the diffusion error for x from the expert π_e and maximize for x generated samples from the policy π_θ. Therefore, $Diff_\phi(x,\epsilon,t)$ will learn to distinguish between expert demonstrations and policy yield samples. π_θ is a generator to generate state-action pairs similar to expert data, confusing the discriminator.

For outer minimization of Eq.(17), it is usually to optimize π_θ by an arbitrary reinforcement learning algorithm with reward signal. So next, we use a diffusion-based discriminator to obtain our surrogate reward function like Gail (Ho and Ermon 2016):

$$R_\phi(x,\epsilon) = -\frac{1}{T}\sum_{t=1}^{T}\log(1-\exp\left(-Diff_\phi(x,\epsilon,t)\right)). \quad (18)$$

During the training process for the diffusion loss, we perform uniform sampling from the interval $[1, T]$. When using diffusion loss as part of the surrogate reward function, we only need to compute the average loss of the diffusion model over N timesteps instead of uniform sampling. We give high rewards to the x with low diffusion errors and vice versa. π_θ can conduct policy improvement from any forward reinforcement learning algorithms with the surrogate reward function. We summarize the overall process of the method in Algorithm 1.

Algorithm 1: Diffusion Adversarial Imitation Learning

Require: initial policy network with parameter θ, Q network with parameter ω and diffusion discriminator network with parameter ϕ, expert data $\mathcal{R}_e$, empty replay buffer $\mathcal{R}$, batch size k, total timesteps N, diffusion timesteps T, learning rate $\eta_\phi, \eta_\theta, \eta_\omega$ for parameters ϕ, θ, ω respectively.

1: let $n = 0$.
2: **while** $n < N$ **do**
3: Collect (s, a, s') according to policy π_θ during interaction and add samples into $\mathcal{R}$.
4: Sample $\{(s_i^{\pi_\theta}, a_i^{\pi_\theta})\}_{i=1}^k \sim \mathcal{R}$, $\{(s_i^{\pi_e}, a_i^{\pi_e})\}_{i=1}^k \sim \mathcal{R}_e$.
5: Sample $\{(t_i)\}_{i=1}^k \sim \mu(1,T), \{(\epsilon_i)\}_{i=1}^k \sim \mathcal{N}(0,I)$.
6: Compute the gradient of diffusion discriminator ϕ:

$$\nabla\mathcal{L} = \nabla_\phi\log(\exp(-Diff_\phi(s_i^{\pi_e}, a_i^{\pi_e}, \epsilon_i, t_i))) + \nabla_\phi\log(1-\exp(-Diff_\phi(s_i^{\pi_\theta}, a_i^{\pi_\theta}, \epsilon_i, t_i))).$$

7: Update diffusion discriminator ϕ by gradient ascent: $\phi \leftarrow \phi + \eta_\phi\nabla\mathcal{L}$
8: Sample $\{(s_i^{\pi_\theta}, a_i^{\pi_\theta})\}_{i=1}^k \sim \mathcal{R}, \{(\epsilon_i)\}_{i=1}^k \sim \mathcal{N}(0,I)$.
9: Compute the surrogate reward $R_\phi(s_i^{\pi_\theta}, a_i^{\pi_\theta}, \epsilon_i)$ by Eq. (18).
10: Update SAC parameters π_θ and Q_ω by η_ϕ and η_ω with the surrogate reward.
11: **end while**

Related Work

Adversarial Imitation Learning

AIL can be viewed as a unified perspective to minimize f-divergences, including Gail (Ho, Jain, and Abbeel 2020), AIRL (Fu, Luo, and Levine 2017), FAIRL (Ghasemipour, Zemel, and Gu 2020), f-Gail (Zhang et al. 2020), etc. These methods aim to minimize divergence by matching the occupancy measure of expert demonstrations and policy data. The policy improvement process and the discriminator optimization process are similar to the GAN (Goodfellow et al. 2014). However, these methods typically rely on on-policy sampling and often require millions or even tens of millions of interactions. To address this issue, DAC (Kostrikov et al. 2018) extends Gail to the off-policy algorithm, significantly improving the learning efficiency of AIL methods and reducing the necessary online interaction to hundreds of thousands of steps. AEAIL (Zhang et al. 2022) proposes to use the reconstruction loss of Autoencoder as the reward function for AIL.

Additionally, a novel category of AIL methods, such as

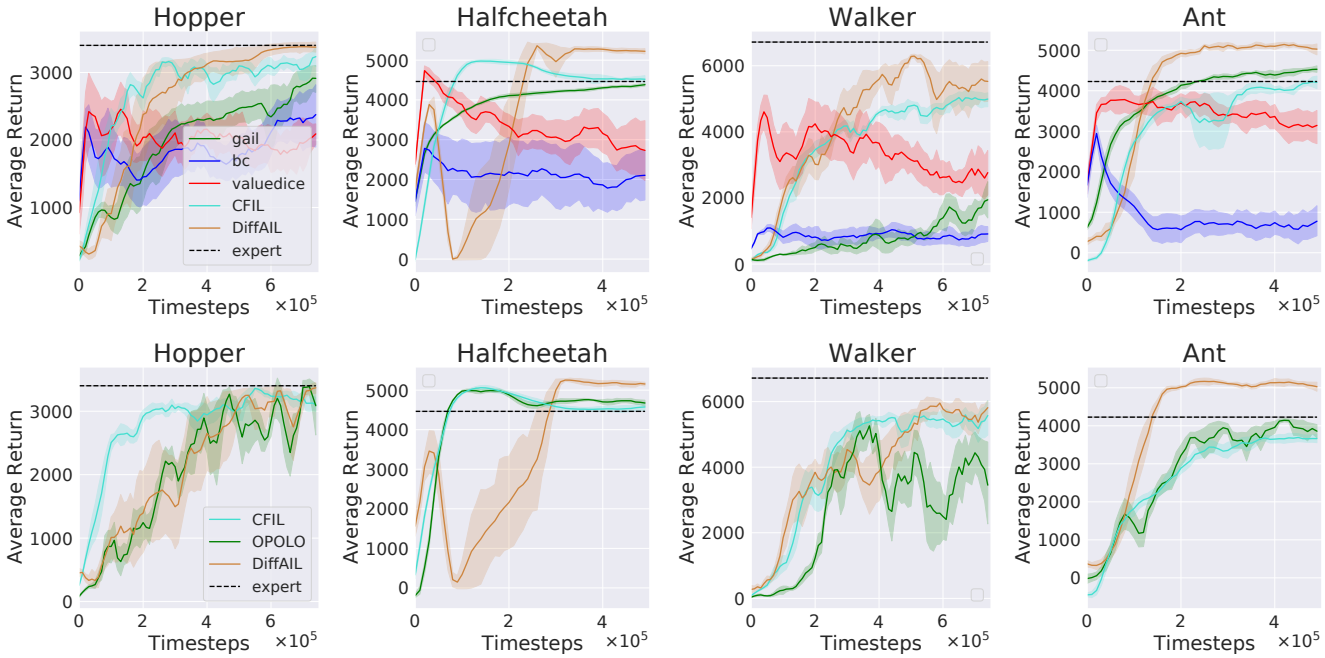

Figure 1: Top: Learning curve for different sota imitation learning algorithms with 1 trajectory 5 five seeds in the standard state-action setting. Bottom: Learning curve for different sota imitation learning algorithms with one trajectory over five seeds in the state-only setting. The x-axis denotes timesteps, and the y-axis denotes the average return. The shadow areas represent the standard deviation.

Valuedice (Kostrikov, Nachum, and Tompson 2019) and CFIL (Freund, Sarafian, and Kraus 2023), employ distribution correction estimation (Nachum et al. 2019) to derive the objective function. They recover the log-density ratio by solving the optimal point of the Donsker-Varadhan representation of the KL divergence. Valuedice builds a fully off-policy optimization objective and outperforms BC in a strictly offline setting. However, recent work (Li et al. 2022) proved that BC is optimal in the strictly offline setting, which raises doubts about the performance of Valuedice. CFIL argues that the log-density ratio cannot be accurately estimated without appropriate modelling tools and proposes using a coupled normalization flow to model the log-likelihood ratio explicitly.

Diffusion Model With Reinforcement Learning

In RL, many methods leverage the powerful generative capabilities of diffusion models to generate state-action pairs or trajectories. Diffusion-BC (Pearce et al. 2023) addresses the limitation of insufficient expressiveness in traditional behavior cloning by modeling diffusion model as a policy. Diffuser (Janner et al. 2022) uses an unconditional diffusion model to generate trajectories consisting of states and actions but requires an additional reward function to guide the reverse denoising process towards high-return trajectories. Diffuser Decision (Ajay et al. 2022) employs a conditional diffusion model with classifier-free guidance to model trajectories containing only states. However, since the modelling process does not involve actions, it is necessary to additionally learn an additional inverse dynamics model to select actions for any two adjacent states along the planned trajectory. BESO (Reuss et al. 2023) applies the conditional diffusion model to goal-conditional imitation learning by directly modelling state and action as conditional distributions and using future states as conditions. Diffusion Q (Wang, Hunt, and Zhou 2022) represents the diffusion model as a policy, explicitly regularizes it, and adds the maximum action value function term to the training objective to adjust the diffusion policy to select actions with the highest Q value.

Our method also utilizes the diffusion model, which differs from the abovementioned diffusion-based methods. Specifically, our approach remains within the AIL framework, unlike BESO, Diffuser Decision, and Diffuser, which directly employ the diffusion model as an expression policy rather than in standard reinforcement learning or imitation learning. Moreover, DiffAIL models an unconditional joint distribution that consists of state-action pairs instead of the conditional distribution.

Experiments

In our experimental section, we explore the performance of the diffusion discriminator on various tasks. Our experiments focus on four key aspects:

- With the popular Mujoco environment, can our method achieve superior results with a small amount of demonstration data in state-action and state-only settings?
- On those expert-level unseen state-action pairs, can Dif-

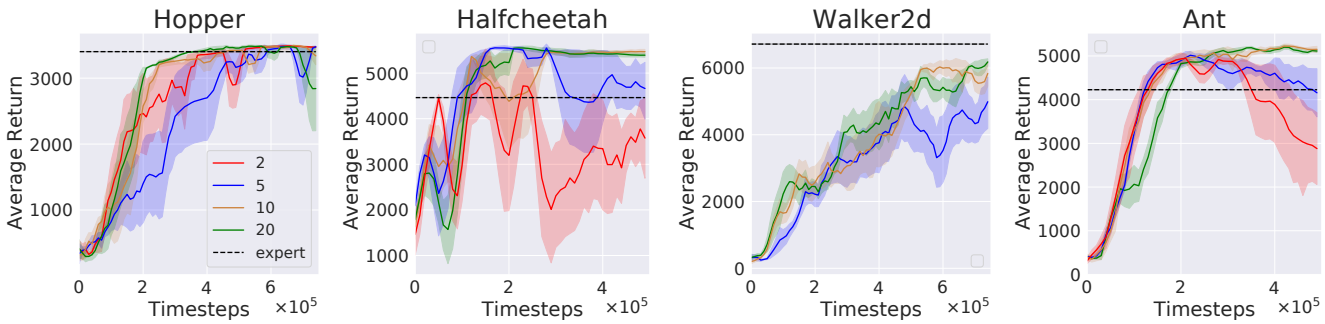

Figure 2: Ablation study on the diffusion timestamps of the diffusion discriminator using N grids [2, 5, 10, 20] with 4 trajectories over 5 seeds. Our findings suggest that a diffusion step of 10 yields good results for the diffusion discriminator. Note that at timestep 2, Walker2d did not show results for some random seeds due to training collapse.

fAIL improve the generalization of successful identification for the discriminator compared to a naïve discriminator?

- Compared to naïve Gail, can DiffAIL provide a more linearly correlational surrogate reward?
- Since the number of diffusion steps significantly impacts model capability, will this phenomenon also occur in our methods?

Benchmark & Dataset. In our experiments, we compare DiffAIL with the SOTA algorithm in the Mujoco environments. We choose four representative tasks, namely Hopper, HalfCheetah, Walker2d, and Ant. We have 40 trajectories on each task dataset and randomly subsample $n = [1, 4, 16]$ trajectories from the pool of 40 trajectories.

Baselines. We benchmark our method against state-of-the-art imitation learning algorithms, including **(1)** Behavior cloning (Pomerleau 1991), the typical supervised learning approach; **(2)** Gail (Ho and Ermon 2016), the classical AIL method; **(3)** Valuedice (Kostrikov, Nachum, and Tompson 2019) which utilize distribution correction estimation; **(4)** CFIL (Freund, Sarafian, and Kraus 2023), which employs a coupled flow to evaluate the log-density ratio both used in state-action and state-only settings; **(5)** OPOLO (Zhu et al. 2020) adjusts policy updates by using a reverse dynamic model to accelerate learning for the state-only setting.

For a fair comparison, we implemented an off-policy version of Gail to improve its sample efficiency, but it did not involve special processing of the bias of the absorbed state reward function in DAC (Kostrikov et al. 2018). Experiments use five random seeds $(0, 1, 2, 3, 4)$, run for an equal number of steps per task, with 10 thousand steps per epoch. Model performance is evaluated every 10 thousand steps by averaging rewards over ten episodes. We then smooth the results using a sliding window of 4 before plotting means and standard deviations across the 5 seeds.

Comparison To SOTA Methods

We conduct experiments on multiple tasks from the popular MuJoCo benchmark, using 1, 4, and 16 expert trajectories to evaluate performance across varying trajectory numbers. For each algorithm, we assess the real average episodic return the agent achieves when interacting in the environment.

With only 1 expert trajectory, the top of Figure 1 shows our method outperforms other imitation learning baselines. Our approach achieves near expert-level asymptotic performance on all tasks and even significantly surpasses the expert on HalfCheetah and Ant. This expert-surpassing result is not performed by other methods. Especially table 1 clearly shows that our method outperforms all other methods significantly on best performance during training.

Subsequently, we conducted our experiment under a state-only setting and found that the advantages brought by the powerful discriminator are very significant, not just in modeling state-action distribution. Under the state-only setting, DiffAIL models the state and the next state distribution. Importantly, we don't need to modify any hyperparameters, and then we directly validate the performance of our method. We compare with OPPLO and CFIL. From the bottom of Figure 1, it can be seen that our method can reach expert-level performance with 1 trajectory and outperform other methods, especially on HalfCheetah and Ant. All of these significant advantages come from the diffusion discriminator being able to accurately capture the data distribution of experts, whether it is state action pairs or state next state pairs.

While our method achieves expert-level performance on multiple tasks, it is less time-efficient than Valuedice's optimized implementation. We acknowledge Valuedice's impressive efficiency gains, though algorithmic speed is not our primary focus. However, in one trajectory situation, we find it can't maintain optimality in later training stages stably. Still, our approach performs similarly or even more efficiently compared to other methods, except for Valuedice.

The experimental results of training additional 4 and 16 trajectories can be seen in Appendix E.

Discriminate On Unused Expert Trajectories

We conduct an additional experiment to verify that our method improves the generalization of unseen expert-level state-action pairs for discrimination. We take the discriminator which has completed training on just 4 trajectories, and test it on 36 held-out expert trajectories from the remaining

Env	Gail	DiffAIL
Hopper	83.3% ± 1.2%	**92.8% ± 2.5%**
HalfCheetah	83.8% ± 0.9%	**94.5% ± 0.6%**
Walker2d	90.5% ± 0.8%	**96.1% ± 0.6%**
Ant	81.7% ± 1.6%	**98.8% ± 0.1%**

Table 2: A comparison of the ability to distinguish expert demonstrations out of data for Gail and DiffAIL with four trajectories over five seeds. It suggests that DiffAIL has a significant improvement over Gail's discriminator.

dataset. If the discriminator can still successfully identify these new trajectories as expert-like, it indicates improved generalization. However, selecting an evaluation metric is challenging. Theoretically, the discriminator will eventually converge to 0.5(Goodfellow et al. 2014), where it cannot distinguish between expert demonstrations and policy data. But in practice, training ends before the discriminator and generator reach the ideal optimum, and the discriminator often overfits the expert demonstrations. Therefore, we adopted a simple evaluation criterion: if the discriminator output is greater than or equal to 0.5, the expert data is considered successfully distinguished; otherwise, the discrimination fails.

As shown in Table 2, it represents the percentage of state-action pairs successfully distinguished by the discriminator from 36 trajectories, i.e., 36000 state-action pairs. From the benchmark of HalfCheetah and Ant, DiffAIL can significantly improve the discrimination generalization of unseen expert demonstrations. In other tasks, DiffAIL also exhibits stronger discriminative abilities than Gail, which sufficiently exhibits the powerful advantage of the diffusion discriminator in capturing the expert distribution. Notably, all trajectories of the same task are yielded by interactions between the same trained policy. Therefore, substantial similarities may exist across state-action pairs from different trajectories, which also explains why Gail can achieve decent discrimination as well.

Return Correlations

As Eq. (3) shows, a better discriminator can provide better surrogate rewards to guide forward reinforcement learning. Therefore, we anticipate that DiffAIL will possess a more positive relevant reward function compared to GAIL. We verify this by experimentally analyzing the Pearson coefficient(PC) between the surrogate discriminator return and the actual return. We compare ours with the vanilla discriminator in GAIL, as shown in Figure 3: The PC for DiffAIL are $[97.1\%, 98.8\%, 99.1\%, 99.6\%]$, while for GAIL are $[97.0\%, 54.1\%, -89.0\%, 92.1\%]$. This indicates that our diffusion surrogate reward function is more linearly correlated with actual rewards than GAIL, which can better guide policy learning.

Ablation Study

We conduct ablation studies to investigate how various diffusion steps impact the distribution-matching ability of the diffusion discriminator on state-action pairs. Figure 2 shows

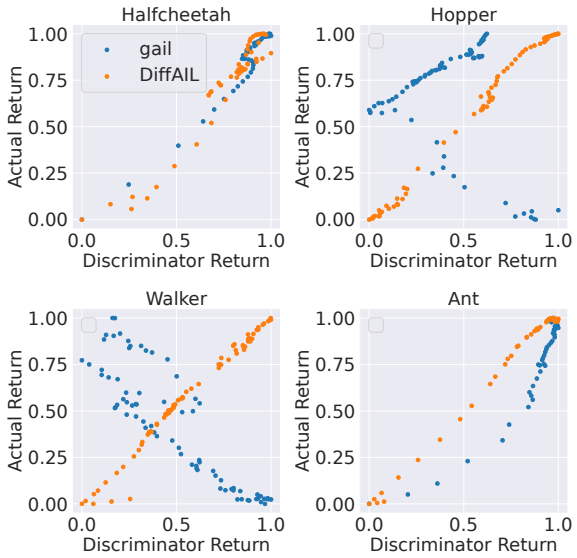

Figure 3: Correlation between average discriminator return and average actual return with Gail and DiffAIL in 3 seeds. The returns have been normalized for display convenience.

that we set the diffusion steps to $t = [2, 5, 10, 20]$, respectively. The results show that increasing diffusion steps leads to more stable performance and faster convergence across all tasks, aligning with results in diffusion models for image generation. However, we also observe that more diffusion steps increase training time, as exhibited in Appendix (Table 4). Therefore, we finally select $t = 10$ as the diffusion step to balance discriminator quality and training cost.

Conclusion

In this work, we propose DiffAIL, which utilizes diffusion models for AIL. Unlike prior methods that directly apply diffusion models as an expressive policy, our approach leverages the diffusion model to enhance the distribution matching ability of the discriminator, which enables the discriminator to improve generalization for expert-level state-action pairs. In both the state-action setting and state-only setting, We model the joint distribution of state-action pairs and state and next state pairs, respectively, with an unconditional diffusion model. Surprisingly, in both settings with a single expert trajectory, the result shows our method achieves SOTA performance across all tasks. Additionally, we discuss the limitations of this work and some promising future directions in Appendix D.

Acknowledgments

This work was supported by the National Science Foundation of China (Nos. 62206159, 62176139), the Natural Science Foundation of Shandong Province (Nos. ZR2022QF117, ZR2021ZD15), the Fundamental Research Funds of Shandong University, the Fundamental Research Funds for the Central Universities. G. Wu was sponsored by the TaiShan Scholars Program.

References

Ajay, A.; Du, Y.; Gupta, A.; Tenenbaum, J.; Jaakkola, T.; and Agrawal, P. 2022. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*.

Freund, G.; Sarafian, E.; and Kraus, S. 2023. A Coupled Flow Approach to Imitation Learning. *arXiv preprint arXiv:2305.00303*.

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

Fujimoto, S.; Hoof, H.; and Meger, D. 2018. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, 1587–1596. PMLR.

Garg, D.; Chakraborty, S.; Cundy, C.; Song, J.; and Ermon, S. 2021. Iq-learn: Inverse soft-q learning for imitation. *Advances in Neural Information Processing Systems*, 34: 4028–4039.

Ghasemipour, S. K. S.; Zemel, R.; and Gu, S. 2020. A divergence minimization perspective on imitation learning methods. In *Conference on Robot Learning*, 1259–1277. PMLR.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, 1861–1870. PMLR.

Ho, J.; and Ermon, S. 2016. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33: 6840–6851.

Janner, M.; Du, Y.; Tenenbaum, J. B.; and Levine, S. 2022. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Kostrikov, I.; Agrawal, K. K.; Dwibedi, D.; Levine, S.; and Tompson, J. 2018. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. *arXiv preprint arXiv:1809.02925*.

Kostrikov, I.; Nachum, O.; and Tompson, J. 2019. Imitation learning via off-policy distribution matching. *arXiv preprint arXiv:1912.05032*.

Li, Z.; Xu, T.; Yu, Y.; and Luo, Z.-Q. 2022. Rethinking ValueDice: Does It Really Improve Performance? *arXiv preprint arXiv:2202.02468*.

Liu, X.-H.; Xu, F.; Zhang, X.; Liu, T.; Jiang, S.; Chen, R.; Zhang, Z.; and Yu, Y. 2023. How To Guide Your Learner: Imitation Learning with Active Adaptive Expert Involvement. *arXiv preprint arXiv:2303.02073*.

Lu, C.; Zhou, Y.; Bao, F.; Chen, J.; Li, C.; and Zhu, J. 2022a. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35: 5775–5787.

Lu, C.; Zhou, Y.; Bao, F.; Chen, J.; Li, C.; and Zhu, J. 2022b. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533.

Nachum, O.; Chow, Y.; Dai, B.; and Li, L. 2019. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *Advances in Neural Information Processing Systems*, 32.

Nowozin, S.; Cseke, B.; and Tomioka, R. 2016. f-gan: Training generative neural samplers using variational divergence minimization. *Advances in neural information processing systems*, 29.

Pearce, T.; Rashid, T.; Kanervisto, A.; Bignell, D.; Sun, M.; Georgescu, R.; Macua, S. V.; Tan, S. Z.; Momennejad, I.; Hofmann, K.; et al. 2023. Imitating human behaviour with diffusion models. *arXiv preprint arXiv:2301.10677*.

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

Reuss, M.; Li, M.; Jia, X.; and Lioutikov, R. 2023. Goal-conditioned imitation learning using score-based diffusion policies. *arXiv preprint arXiv:2304.02532*.

Ross, S.; Gordon, G.; and Bagnell, D. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 627–635. JMLR Workshop and Conference Proceedings.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015. Trust region policy optimization. In *International conference on machine learning*, 1889–1897. PMLR.

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

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587): 484–489.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Todorov, E.; Erez, T.; and Tassa, Y. 2012. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, 5026–5033. IEEE.

Wang, Z.; Hunt, J. J.; and Zhou, M. 2022. Diffusion policies as an expressive policy class for offline reinforcement learning. *arXiv preprint arXiv:2208.06193*.

Zhang, K.; Zhao, R.; Zhang, Z.; and Gao, Y. 2022. Auto-Encoding Adversarial Imitation Learning.

Zhang, X.; Li, Y.; Zhang, Z.; and Zhang, Z.-L. 2020. f-gail: Learning f-divergence for generative adversarial imitation learning. *Advances in neural information processing systems*, 33: 12805–12815.

Zheng, K.; Lu, C.; Chen, J.; and Zhu, J. 2023. DPM-Solver-v3: Improved Diffusion ODE Solver with Empirical Model Statistics. *arXiv preprint arXiv:2310.13268*.

Zhu, Z.; Lin, K.; Dai, B.; and Zhou, J. 2020. Off-policy imitation learning from observations. *Advances in Neural Information Processing Systems*, 33: 12402–12413.