INCOMPATIBILITY BETWEEN DETERMINISTIC POLICY AND GENERATIVE ADVERSARIAL IMITATION LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Deterministic policies are widely applied in generative adversarial imitation learning (GAIL). When adopting these policies, some GAIL variants modify the reward function to avoid training instability. However, the mechanism behind this instability is still largely unknown. In this paper, we capture the instability through the underlying exploding gradients theoretically in the updating process. Our novelties lie in the following aspects: 1) By employing multivariate Gaussian policy with small covariance to approximate deterministic policy, we establish and prove the probabilistic lower bound for the exploding gradients, which can describe the degree of instability universally, while the stochastic policy will never suffer from such pathology subsequently. 2) We also prove that the modified reward function of adversarial inverse reinforcement learning (AIRL) can relieve exploding gradients, but at the expense of "non-confidence". Experiments and a toy demo support our analysis.

1 INTRODUCTION

Imitation learning (IL) trains a policy directly from expert demonstrations without reward signals (Ng et al., 2000; Syed & Schapire, 2007; Ho & Ermon, 2016). It has been broadly studied under the twin umbrellas of behavioral cloning (BC) (Pomerleau, 1991) and inverse reinforcement learning (IRL) (Ziebart et al., 2008). Generative adversarial imitation learning (GAIL) (Ho & Ermon, 2016) plugs the inspiration of generative adversarial networks (GANs) (Goodfellow et al., 2014) into maximum entropy IRL. The discriminator in GAIL aims to distinguish whether a state-action pair comes from the expert demonstration or is generated by the agent. Meanwhile, the learned policy generates interaction data for confusing the discriminator. GAIL is promising for many real-world scenarios where designing reward functions to learn the optimal control policies requires significant effort. It has made remarkable achievements in physical-world tasks, e.g., robot manipulation (Jabri, 2021), mobile robot navigating (Tai et al., 2018), commodities search (Shi et al., 2019), endovascular catheterization (Chi et al., 2020), etc.

The learned policy in GAIL can be effectively accomplished by reinforcement learning (RL) methods (Sutton & Barto, 2018; Puterman, 2014), which are divided into stochastic policy algorithms and deterministic policy algorithms, incorporating the two classes of algorithms into GAIL denoted as ST-GAIL and DE-GAIL respectively in our paper. For ST-GAIL, one can refer to the proximal policy optimization (PPO)-GAIL (Chen et al., 2020), the natural policy gradient (NPG)-GAIL (Guan et al., 2021b) and the two-stage stochastic gradient (TSSG) (Zhou et al., 2022). These algorithms have shown that GAIL can ensure global convergence in high-dimensional environments against traditional IRL methods (Ng et al., 2000; Ziebart et al., 2008; Boularias et al., 2011). Unfortunately, ST-GAIL methods have low sample efficiency and cost a long time to train the learned policy well (Zuo et al., 2020).

In comparison, some related works (Kostrikov et al., 2019; Zuo et al., 2020) imply that deterministic policies are capable of enhancing sample efficiency when training GAIL variants. Kostrikov et al. (2019) proposed the discriminator-actor-critic (DAC) algorithm, which defines the reward function with the GAIL discriminator for the policy trained by the twin delayed deep deterministic policy gradient (TD3) (Fujimoto et al., 2018). It reduces policy-environment interaction sample complexity

by an average factor of 10. Deterministic generative adversarial imitation learning (DGAIL) (Zuo et al., 2020) utilizes the modified deep deterministic policy gradient (DDPG) (Lillicrap et al., 2015) to update the policy, and achieves a faster learning speed than ST-GAIL.

These DE-GAIL methods not only effectively improve sample efficiency, but also mitigate instability caused by the reward function $-\log(1 - D(s, a))$, referred to as the *positive logarithmic reward function* (PLR). This classical logarithmic reward function is one of the primary shapes and is frequently used in GAIL. For PLR, Kostrikov et al. (2019) discussed the reward bias in a simple toy demo; Zuo et al. (2020) exhibited its instability through contrast experiments, and improved the stability of DE-GAIL by introducing the idea of learning from demonstrations (LfD) into the generator.

However, the instability caused by PLR is largely unknown. We investigate the performance of TD3-GAIL with 10^6 expert demonstrations displayed in Fig. 1. It emerges extreme irregularity under nine random seeds. The learning effects under some seeds are capable of reaching expert levels (valid), while others hardly learn anything (invalid). At some point, the two corner cases can not be well captured by the bias analysis introduced through the toy demo in Kostrikov et al. (2019).

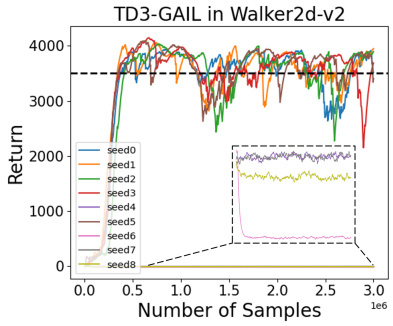


Figure 1: Learning curves of TD3-GAIL with different random seeds in Walker2d-v2.

To reveal the principle of such instability more reasonably, first, we incorporate a different deterministic policy algorithm, i.e., the softmax deep double deterministic policy gradients (SD3) (Pan et al., 2020) into GAIL. SD3-GAIL exhibits the same irregularity as the experiment results in TD3-GAIL and DGAIL. This implies that instability is not a special case in DE-GAIL. In addition, the instability of DE-GAIL is mainly down to the invalid cases that appeared in the experiment. A toy demo is designed to explore this invalidity. In this demo, PLR in DE-GAIL tends to zero as the state space size approaches infinity. Then, we prove that there exist exploding gradients in the objective of policy training, describing the invalidity theoretically and universally. Meanwhile, we give the probabilistic lower bound for exploding gradients, with respect to the mismatching between the learned policy and the expert demonstration. In this way, the possibility of both valid and invalid cases can be better clarified. Finally, we disclose that the "aggressive interval" under the modified reward function in adversarial inverse reinforcement learning (AIRL) (Fu et al., 2018) is smaller than that under PLR in GAIL. Such a modified reward function shows its superiority of stability but at the cost of "non-confidence" in DE-GAIL. Our contributions can be summarized as follows:

- Establish and prove the probability of exploding gradients (PEG) theorem to theoretically describe the instability in DE-GAIL. In contrast, our analysis strikes that stochastic policy can avoid exploding gradients.
- A toy demo and experiments in three MuJoCo environments validate our analysis. Compared with the consistent trend in ST-GAIL, we point out the instability is caused by deterministic policies, rather than GANs.
- The reward function in AIRL is shown to reduce the probability of exploding gradients while at the risk of "non-confidence".

2 Related Work

In large and high-dimensional environments, Ho & Ermon (2016) proposed GAIL, which is processed on the trust region policy optimization (TRPO) (Schulman et al., 2015). It gains significant performance in imitating complex expert policies (Ghasemipour et al., 2019; Xu et al., 2020; Ke et al., 2020; Chen et al., 2021).

To accelerate the GAIL learning process, a natural idea is using deterministic policy gradients. Sample-efficient adversarial mimic (SAM) (Blondé & Kalousis, 2019) method integrates DDPG

(Lillicrap et al., 2015) into GAIL, and adds a penalty on the gradient of the discriminator meanwhile. Zuo et al. (2020) proposed deterministic generative adversarial imitation learning (DGAIL), which combines the modified DDPG with LfD to train the generator under the guidance of the discriminator. The reward function in DGAIL is set as $D(s, a)$. TD3 (Fujimoto et al., 2018) and off-policy training of the discriminator are performed in DAC (Kostrikov et al., 2019) to reduce policy-environment interaction sample complexity by an average factor of 10. The revised reward function in DAC is $\log(D(s, a)) - \log(1 - D(s, a))$.

Notably, these works achieve gratifying results benefiting from modifications to reward functions. When implementing GAIL with PLR directly, Kostrikov et al. (2019) pointed out the reward bias via a special toy demo; DGAIL exhibited its instability merely from the perspective of experimental results. Differently, we illustrate this phenomenon through a universal theory.

3 PRELIMINARY

In this section, we introduce the definition of a Markov decision process, reproducing kernel Hilbert space and generative adversarial imitation learning setup.

3.1 MARKOV DECISION PROCESS

A discounted Markov decision process (MDP) is characterized by a 5-tuple $(\mathcal{S}, \mathcal{A}, r, p_M, \gamma)$ in the standard RL setting. $\mathcal{S}$ and $\mathcal{A}$ denote the finite state space and action space, respectively. $r(s, a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function for performing action $a \in \mathcal{A}$ in state $s \in \mathcal{S}$. $p_M(s'|s, a) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ denotes the transition distribution and γ is the discount factor. A policy $\pi(a|s)$ specifies an action distribution conditioned on state s. The objective of RL is to maximize the expected reward-to-go $\eta(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t, a_t\right) | s_0, a_0\right]$.

Induced by a policy π, we define *the discounted stationary state distribution* as

$$d^\pi(s) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \mathrm{Pr}(s_t = s; \pi).$$

Here $\mathrm{Pr}(s_t = s; \pi)$ denotes the probability of reaching state s at time t, which is given by

$$\int \prod_{u=0}^{t-1} p_M\left(s_{u+1} \mid s_u, a_u\right) \pi\left(a_u \mid s_u\right) \mathrm{Pr}\left(s_0\right) d\boldsymbol{s} d\boldsymbol{a},$$

where $d\boldsymbol{s} = ds_0 \ldots ds_{t-1}$ and $d\boldsymbol{a} = da_0 \ldots da_{t-1}$ imply integration over the previous states and actions. Similarly, *the discounted stationary state-action distribution* is defined as

$$\rho^\pi(s, a) = (1 - \gamma) \sum_{t=0}^{\infty} \gamma^t \mathrm{Pr}(s_t = s, a_t = a; \pi),$$

which measures the overall "frequency" of visiting a state-action pair under the policy π. The relationship between $\rho^\pi(s, a)$ and $d^\pi(s)$ can be described as

$$\rho^\pi(s, a) = \pi(a|s) d^\pi(s). \tag{1}$$

3.2 REPRODUCING KERNEL HILBERT SPACE

Given a set $\mathcal{S}$, for any $c \in \mathbb{R}^p$ and $x \in \mathcal{S}$, if the linear functional mapping $h \in \mathcal{H}$ to $(c, h(x))$ is continuous, then a vector-valued reproducing kernel Hilbert space (RKHS) $\mathcal{H}$ is a Hilbert space of functions $h : \mathcal{S} \to \mathbb{R}^p$ (Micchelli & Pontil, 2005). For all $y \in \mathcal{S}$, $\kappa_x(y)$ is a symmetric function that is a positive definite matrix, thereby $(\kappa_x c)(y) = \kappa(x, y)c \in \mathcal{H}$. It has the reproducing property $\langle h, \kappa_x c\rangle_{\mathcal{H}} = h(x)^\top c$. Here, $(\cdot, \cdot)$ and $\langle \cdot, \cdot\rangle_{\mathcal{H}}$ denote the inner product in $\mathbb{R}^p$ and in $\mathcal{H}$, respectively.

3.3 GENERATIVE ADVERSARIAL IMITATION LEARNING

GAIL combines IRL with GANs, treating RL methods as the generator. GAIL takes advantage of the discriminator $D(s, a)$ to calculate the difference between the distribution of the state-action

pair induced by the learned policy π and the expert policy π_{E}, thereby providing the reward for the agent. Moreover, the policy and discriminator can be approximated by RKHS (Ormoneit & Sen, 2002). The optimization problem in GAIL is

$$\min_{\pi} \max_{D} \mathbb{E}_{(s,a)\sim\rho^{\pi_{\mathrm{E}}}}[\log(D(s,a))] + \mathbb{E}_{(s,a)\sim\rho^{\pi}}[\log(1-D(s,a))], \tag{2}$$

where the policy π mimics the expert policy via the reward function $r(s,a) = -\log(1-D(s,a))$. When the discriminator reaches its optimal

$$D^*(s,a) = \rho^{\pi_{\mathrm{E}}}(s,a)/(\rho^{\pi_{\mathrm{E}}}(s,a) + \rho^{\pi}(s,a)), \tag{3}$$

the optimization objective of the learned policy is formalized as minimizing the state-action distribution discrepancy between the imitated policy and the expert policy with the Jensen-Shannon (JS) divergence:

$$\min_{\pi} D_{\mathrm{JS}}(\rho^{\pi}(s,a), \rho^{\pi_{\mathrm{E}}}(s,a)) := \frac{1}{2} D_{\mathrm{KL}}\left(\rho^{\pi}, \frac{\rho^{\pi}+\rho^{\pi_{\mathrm{E}}}}{2}\right) + \frac{1}{2} D_{\mathrm{KL}}\left(\rho^{\pi_{\mathrm{E}}}, \frac{\rho^{\pi}+\rho^{\pi_{\mathrm{E}}}}{2}\right). \tag{4}$$

4 The principle of instability in GAIL with Deterministic Policies

We first show the performance of DE-GAIL in Subsect. 4.1. Inspired by a toy demo designed to describe the invalidity of DE-GAIL in Subsect. 4.2, we theoretically impute the invalidity to exploding gradients in the objective of policy training in Subsect. 4.3. Additionally, the probabilistic lower bound for exploding gradients is provided in Theorem 1. In contrast, we present the compatibility between stochastic policy and GAIL. Finally, we prove that the reward function in AIRL relieves the exploding gradients at the cost of "non-confidence" in Subsect. 4.4.

4.1 Illness Reward Empirical Evaluations in MuJoCo environments

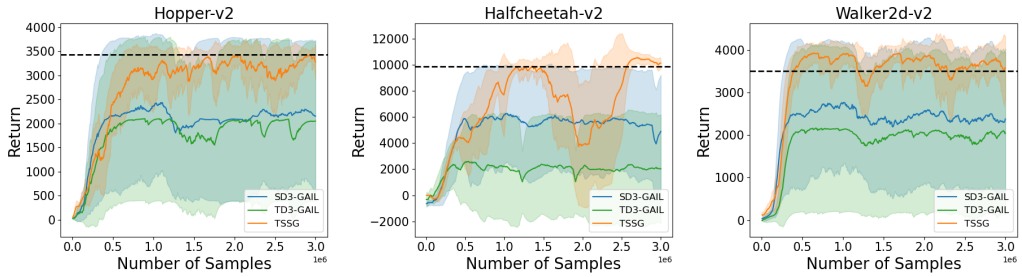


Figure 2: Comparison of SD3-GAIL, TD3-GAIL and TSSG in three different environments. Solid lines and dashed lines correspond to the average performance of the three algorithms and expert demonstrations, respectively.

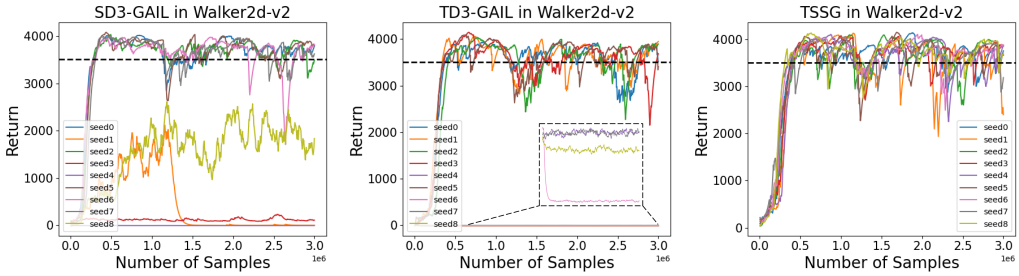
We replicate the experimental setup of Zhou et al. (2022): expert trajectories are created by SAC agent in Hopper-v2, HalfCheetah-v2 and Walker2d-v2 respectively. The size of expert demonstration data is 10^6 obtained with a 0.01 standard deviation. The mean return of the demonstration data in each environment is 3433, 9890 and 3509 respectively.

Algorithm 1 GAIL with deterministic policy algorithms

1: **Input:** Expert demonstrations, initialize the learned policy π_θ and discriminator network D, some RL deterministic policy algorithms, such as SD3 and TD3.
2: **for** iteration $0, 1, 2, \cdots$ **do**
3: Update D by maximizing $\mathbb{E}_{(s,a)\sim\rho^{\pi_{\mathrm{E}}}}[\log(D(s,a))] + \mathbb{E}_{(s,a)\sim\rho^{\pi}}[\log(1-D(s,a))]$.
4: Calculate PLR $r(s,a) = -\log(1-D(s,a))$.
5: Update θ by a deterministic policy algorithm.
6: **end for**

When training GAIL, we use two-layer networks to approximate the kernel function (see Arora et al. (2019)). The reward function is set as $r(s,a) = -\log(1-D(s,a))$ (PLR). First, a well-performed

TSSG is revisited as the baseline for subsequent comparison. Then we explore TD3, which also causes GAIL to fail as Kostrikov et al. (2019). A brief description of the training procedure is laid out in the algorithm pseudocode (Alg. 1). The best evaluation results of TD3-GAIL are shown in Fig. 2. We can observe that compared to TSSG, TD3-GAIL not only obtains a lower mean return but also suffers from obviously higher variance. We suspect this is owing to that TD3 leading to a large Q-value underestimation (Pan et al., 2020).



(a) SD3-GAIL (b) TD3-GAIL (c) TSSG

Figure 3: The performances of the three algorithms under different 9 seeds in Walker2d-v2.

We next experimentally examine whether better value estimation will improve GAIL. SD3 (Pan et al., 2020), a deterministic policy algorithm that enables a smaller absolute bias of true values and value estimates than TD3, is conducted into the framework of GAIL as shown in Alg. 1. Its best performances are presented in Fig. 2. We conclude the results in two aspects:

- SD3-GAIL improves the average return compared to TD3-GAIL, and even achieves a higher upper limit than TSSG. This phenomenon is possibly attributed to the accuracy of Q-value estimation in SD3.
- Although the enhancement of average return in SD3-GAIL compared to TD3-GAIL, the variance of SD3-GAIL remains high. In order to thoroughly observe the phenomenon of high variance of SD3-GAIL and TD3-GAIL, we specify the performance of the three algorithms under 9 random seeds in Walker2d-v2 as exhibited in Fig. 3. The two deterministic policy algorithms both reveal extreme irregularity under nine random seeds. Specifically,
  - The performances under some seeds are capable of reaching expert levels, such as SD3-GAIL and TD3-GAIL with seed 5.
  - Others, such as SD3-GAIL with seed 3 and TD3-GAIL with seed 8 almost learn nothing.
  - SD3-GAIL with seed 1 and seed 8 learn only part of the expert policy.

  While TSSG maintains a similar trend under different random seeds, which manifests that the instability is not caused by GANs itself depicted by Arjovsky & Bottou (2017).

These empirical evaluations manifest that DE-GAIL algorithms suffer from pathological training dynamics. We attempt to explain the experimental phenomenon by employing the statement of reward bias introduced by Kostrikov et al. (2019). In their specific toy demo, repeated trajectories may be trained with the stochastic learned policy, which brings about a higher return than expert under PLR $r(s,a) = -\log(1 - D(s,a))$, the strictly positive reward function. This is known as reward bias. Conversely, if the learned policy is deterministic, the trajectories will be trained with no repetition. Furthermore, the expert return will not be exceeded by the return of the deterministic learned policy technically, in other words, the reward bias vanishes. This can not interpret DE-GAIL as inferior to ST-GAIL that appeared in our experiment. In the sequel, we will explain such instability from the perspective of invalidity in DE-GAIL.

4.2 The invalidity in DE-GAIL

The instability of DE-GAIL is mainly imputed to the invalidity case, i.e., the reward function approaches zero. To illustrate this invalidity, consider a specific MDP $\mathcal{M}$: $\mathcal{S}$ contains $(g+1)$ states

$s_0, s_1, ..., s_g$, where s_g is the terminal state. The action $a_{i\to j}$ in $\mathcal{A}$ is such that $s_j \sim p(\cdot|s_i, a_{i\to j})$ satisfies that $|i-j| \le g/4,\ i \ne j$. The expert demonstration is shown in Fig. 4.

For any state s_t, the action $a_{t\to t+1}$ occurs 3 times while others occur at most twice in the expert demonstration. Therefore, the most frequent action $a_{t\to t+1}$ is the best choice for a deterministic learned policy under each s_t. The detailed trajectory is shown in Fig. 5.

We now show the behavior of rewards for state-action pairs in the deterministic policy.

Proposition 1 *In the MDP $\mathcal{M}$, under the optimal discriminator, PLR of $(s_t, a_{t\to t+1})$ satisfies*

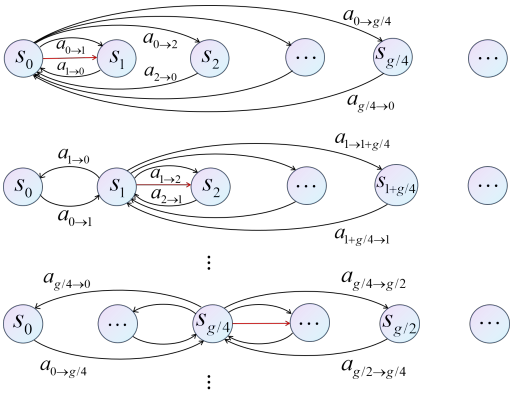
$$r(s_t, a_{t\to t+1}) = -\log(1 - D(s_t, a_{t\to t+1})) \to 0 \quad as \quad |\mathcal{S}| \to \infty,$$

where

$$D(s_t, a_{t\to t+1}) = \begin{cases} \frac{48g}{11g^2+72g-16}, & 0 \le t < \frac{3g}{4}, \\ \frac{16g}{11g^2+40g-16}, & t = \frac{3g}{4}, \\ 0, & \text{otherwise.} \end{cases}$$

Proof. See Appendix A.1. □

Remark 1 *In contrast, for a stochastic policy reaching the expert level, the length of the expert trajectory and the stochastic policy have the same order $\mathcal{O}(|\mathcal{S}|^2)$. Here $|\mathcal{S}|$ denotes the size of $\mathcal{S}$. Referring to the proof strategy in Appendix A.1, PLR will not decline to zero. Proposition 1 illustrates the invalid case in DE-GAIL, demonstrating the incompatibility between deterministic policy and GAIL.*



The expert demonstration in the designed demo

Input: $\mathcal{S} = [s_0, \cdots, s_g]$
for $i = 0, \cdots, 3g/4 - 1$
  for $|i-j| \le g/4, j \ge 0, j \ne i$
    Take the actions $a_{i\to j}$ and $a_{j\to i}$.
  end for
  Take the action $a_{i\to(i+1)}$.
end for
for $j = g/2, \cdots, 3g/4-1, 3g/4+1, \cdots, g-1$
  Take the actions $a_{3g/4\to j}$ and $a_{j\to 3g/4}$.
end for
Take the action $a_{3g/4\to g}$.

Figure 4: Expert demonstration. Each state transfers to the neighboring $g/4$ states and back to itself from left to right. **Left:** The schematic of the expert demonstration, where the red arrow is the last action taken in each state. **Right:** The pseudocode of the expert demonstration.

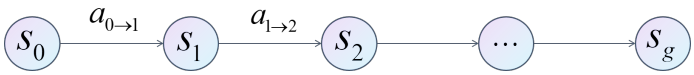


Figure 5: The trajectory of the deterministic learned policy. From s_0 to the terminal state s_g sequentially.

4.3 Exploding gradients in DE-GAIL

Inspired by Lever & Stafford (2015) and Paternain et al. (2020) who employ multivariate Gaussian policy to approximate deterministic policy, we define the learned policy π_h as

$$\pi_h(a|s) = \frac{1}{\sqrt{\det(2\pi\mathbf{\Sigma})}} \exp \frac{-(a-h(s))^\top \mathbf{\Sigma}^{-1}(a-h(s))}{2},$$

with mean vector $h(s)$ and covariance matrix $\mathbf{\Sigma}$. Note that $\pi_h(a|s)$ can be regarded as an approximation to the Dirac's impulse via covariance matrix approaching zero (Paternain et al., 2020), i.e.,

$$\lim_{\mathbf{\Sigma}\to\mathbf{0}} \pi_h(a|s) = \delta(a - h(s)). \tag{5}$$

Eq. (5) means that when the covariance $\mathbf{\Sigma} \to \mathbf{0}$, the stochastic policy $\pi_h(a|s)$ approaches the deterministic policy $h(s)$. Replacing π with π_h in Eq. (2), Eq. (3) and Eq. (4) respectively, the optimization problem of GAIL under π_h is

$$\min_{\pi_h} \max_{D} \ \mathbb{E}_{(s,a)\sim\rho^{\pi_{\mathrm{E}}}}[\log(D(s,a))] + \mathbb{E}_{(s,a)\sim\rho^{\pi_h}}[\log(1 - D(s,a))],$$

the optimum discriminator is

$$D^*(s,a) = \rho^{\pi_{\mathrm{E}}}(s,a)/(\rho^{\pi_{\mathrm{E}}}(s,a) + \rho^{\pi_h}(s,a)), \tag{6}$$

and the policy optimization objective is

$$\min_{\pi_h} D_{\mathrm{JS}}(\rho^{\pi_h}(s,a), \rho^{\pi_{\mathrm{E}}}(s,a)).$$

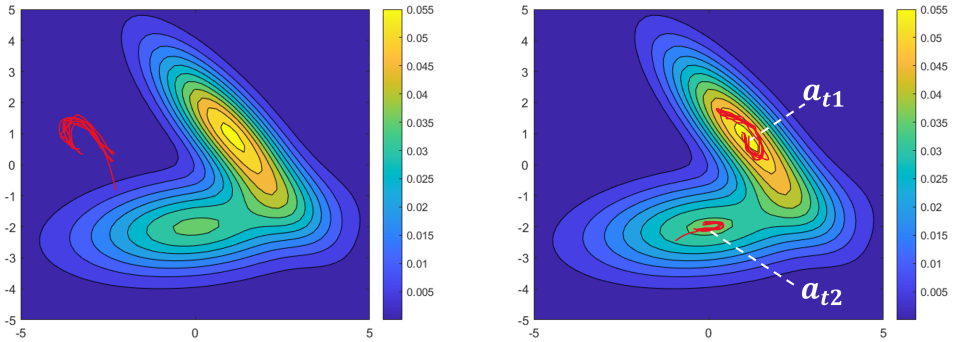


Figure 6: The process of the mismatched case and the matched case. **Left:** The mismatched case. **Right:** The matched case.

Before proceeding with our main result, we need a crucial definition.

Definition 1 (Mismatched and Matched State-action Pair) *The state-action pair $(s_t, h(s_t))$ induced by the learned policy mismatches the expert (s_t, a_t) if $\|h(s_t) - a_t\|_2 \geq C\|\mathbf{\Sigma}\|_2$ for any $C > 0$. Otherwise, $(s_t, h(s_t))$ matches the expert.*

We utilize an event

$$\Xi = \{(s_t, h(s_t)) : \|h(s_t) - a_t\|_2 \geq C\|\mathbf{\Sigma}\|_2 \text{ for any } C > 0\}$$

to characterize the mismatching.

The probability density contour map of the expert action demonstration in state s_t is shown in Fig. 6. The threshold of matching is set as 0.035. We claim the red lines as the training trajectories. The optimal and sub-optimal actions by the expert are denoted as a_{t1} and a_{t2} respectively. More precisely, the optimal matching is that $h(s_t)$ lays in the neighborhood of a_{t1}, which has the most density in the expert demonstration, while the sub-optimal matching means that $h(s_t)$ matches the second intensive a_{t2}. The learned policy is exposure to be transferred to the mismatching case due to the limited area in the sub-optimal matching compared with the optimal; see seed 1 in Fig. 3(a) for instance.

Now we present our main theorem on the probability of exploding gradients (PEG) in GAIL.

Theorem 1 (Probability of Exploding Gradients) *Let $\pi_h(\cdot|s)$ be the Gaussian stochastic policy with mean $h(s)$ and covariance $\mathbf{\Sigma}$. When $D(s,a)$ achieves its optimum in Eq. (6), the gradient estimator of the policy loss with respect to the policy's parameter h is given by*

$$\hat{\nabla}_h D_{\mathrm{JS}}(\rho^{\pi_h}, \rho^{\pi_{\mathrm{E}}}) = \frac{d^{\pi_h}(s_t)\nabla_h \pi_h(a_t|s_t)}{2d^{\pi_{\mathrm{E}}}(s_t)\pi_{\mathrm{E}}(a_t|s_t)} \log \frac{2d^{\pi_h}(s_t)\pi_h(a_t|s_t)}{d^{\pi_h}(s_t)\pi_h(a_t|s_t) + d^{\pi_{\mathrm{E}}}(s_t)\pi_{\mathrm{E}}(a_t|s_t)},$$

where $\nabla_h \pi_h(a|s) = \pi_h(a|s)\kappa(s,\cdot)\mathbf{\Sigma}^{-1}(a - h(s))$. *Then we have*

$$\|\hat{\nabla}_h D_{\mathrm{JS}}(\rho^{\pi_h}, \rho^{\pi_{\mathrm{E}}})\|_2 \to \infty$$

with a probability of $\Pr(\|\mathbf{\Sigma}^{-1}(a_t - h(s_t))\|_2 \geq C$ *for any* $C > 0)$.

Proof. See Appendix A.2. □

Remark 2 *When* $\mathbf{\Sigma} \to \mathbf{0}$*, in other words, the policy is deterministic, we have*

$$\begin{aligned}\Pr(\|\mathbf{\Sigma}^{-1}(a_t - h(s_t))\|_2 \geq C \text{ for any } C > 0) &\geq \Pr(\|a_t - h(s_t)\|_2 \geq C\|\mathbf{\Sigma}\|_2 \text{ for any } C > 0) \\ &= \Pr(\Xi).\end{aligned}$$

$\Pr(\Xi)$ *is nontrivial since* $\mathbf{\Sigma} \to \mathbf{0}$*. It reveals that under the optimal discriminator, deterministic policy in GAIL will suffer from exploding gradients with a probability of at least* $\Pr(\Xi)$*. We conclude that deterministic policy is incompatible with GAIL. Now, the invalid cases aforementioned in DE-GAIL are clearly illustrated.*

In contrast, for a Gaussian stochastic policy (fixed $\mathbf{\Sigma}$*), we have*

$$\Pr(\|\mathbf{\Sigma}^{-1}(a_t - h(s_t))\|_2 \geq C \text{ for any } C > 0) = 0.$$

Thus, under the optimal discriminator, Gaussian stochastic policy in GAIL is guaranteed not to suffer from exploding gradients. Analogous conclusions can be drawn for non-Gaussian stochastic policies.

4.4 Relieving Exploding Gradients with Reward Modification

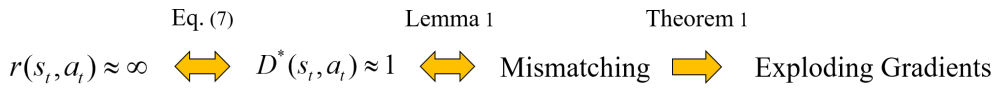


Figure 7: Analysis of exploding gradients.

Kostrikov et al. (2019) introduced the reward function $r_2(s_t, a_t) = \log(D(s_t, a_t)) - \log(1 - D(s_t, a_t))$ of AIRL into GAIL, and illustrated a reduction in instability through comparative experiments against GAIL with PLR $r_1(s_t, a_t) = -\log(1 - D(s_t, a_t))$. The reward function of AIRL is defined as the *combination reward function* (CR) in Wang & Li (2021). For written convenience, DE-GAIL with PLR and CR are called PLR-DE-GAIL and CR-DE-GAIL respectively.

Further, in this section, we proceed with this reduction of instability from a theoretical explanation, in other words, whether CR-DE-GAIL permits a relief in exploding gradient probability over PLR-DE-GAIL.

Theorem 1 points out that the mismatched state-action pair results in exploding gradients. Further, what truly pertains to our discussion in the sequel is the behavior of the discriminator. This is due to its unified characteristic that lies in the interval $[0, 1]$. Now we present the following Lemma on the mismatching in the view of the discriminator.

Lemma 1 *Under the optimal discriminator in Eq. (6), we have*

$$D^*(s_t, a_t) \approx 1 \Leftrightarrow h(s_t) \text{ mismatches } a_t.$$

Proof. The optimal discriminator of (s_t, a_t) can be denoted by

$$D^*(s_t, a_t) = \frac{\rho^{\pi_{\mathrm{E}}}(s_t, a_t)}{\rho^{\pi_{\mathrm{E}}}(s_t, a_t) + \rho^{\pi_h}(s_t, a_t)}.$$

We can derive that the necessary and sufficient condition of $D^*(s_t, a_t) \approx 1$ is that $\rho^{\pi_h}(s_t, a_t) \approx 0$, i.e., $h(s_t)$ mismatches a_t. □

Lemma 1 indicates that exploding gradients depend on the distance between the discriminator's value and 1, or the degree of $r_i(s_t, a_t),\ i = 1, 2$ approaching infinity. This is due to the monotonicity of both $r_1(s_t, a_t)$ and $r_2(s_t, a_t)$, and we thus obtain

$$r_1(s_t, a_t) \approx \infty \text{ and } r_2(s_t, a_t) \approx \infty \tag{7}$$

as $D(s_t, a_t) \approx 1$. The relationship between exploding gradients and $r_i(s_t, a_t) \approx \infty$ is depicted in Fig. 7.

Naturally, to prevent exploding gradients, we make the constraints that $r_i(s_t, a_t) \leq C,\ i = 1, 2$ for some appropriate constant C. According to this, the following definition is presented.

Definition 2 *Under the optimal discriminator in Eq. (6), an "aggressive interval" of the discriminator is defined as* $[\alpha, 1]$ *such that* $r_1(s_t, a_t) \geq C$. *Similarly, under the same upper bound* C, *the "aggressive interval" of the discriminator is* $[\beta, 1]$ *for* $r_2(s_t, a_t)$.

We note that when the discriminator comes to rest in $[\alpha, 1]$, then the training process will suffer from exploding gradients. The remission of exploding gradients in CR-DE-GAIL is presented in the following proposition.

Proposition 2 *Under the optimal discriminator in Eq. (6), we have* $\beta \geq \alpha$.

Proof. When $r_i(s_t, a_t) = C,\ i = 1, 2$, we obtain $\log \beta - \log(1 - \beta) = -\log(1 - \alpha)$, which is followed by

$$\beta - \alpha = \frac{\alpha^2 - 2\alpha + 1}{2 - \alpha} \geq 0.$$

□

Proposition 2 reveals that the discriminator in CR-DE-GAIL exhibits a smaller "aggressive interval" than that in PLR-DE-GAIL, dropping the probability of gradient explosion. On the other hand, for a matched state-action pair, the discriminator in CR-DE-GAIL requires a larger value to reach the same reward as in PLR-DE-GAIL. This is a lack of confidence in the well-trained policy to some extent, and we write it as "non-confidence".

5 Conclusion

In this paper, we have explored the principle of instability in DE-GAIL. We first experimentally show the extremely instability performance of DE-GAIL algorithms compared to ST-GAIL. Subsequently, our theoretical proof manifests that the gradient of the deterministic policy loss with respect to the policy will suffer from exploding with some probability, thereby leading to training failure. In comparison, we present the compatibility between stochastic policy and GAIL. Finally, the modified reward function is shown to remedy the exploding gradients, but at the expense of the "non-confidence".

Our analysis of DE-GAIL focused on the possible exploding gradients. Further analysis of GAIL variants that reduce the "non-confidence" phenomenon while still being non-exploding gradients is left for future works.

References

Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, volume 32, pp. 8139–8148, 2019.

Lionel Blondé and Alexandros Kalousis. Sample-efficient imitation learning via generative adversarial nets. In *International Conference on Artificial Intelligence and Statistics*, pp. 3138–3148, 2019.

Abdeslam Boularias, Jens Kober, and Jan Peters. Relative entropy inverse reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 182–189, 2011.

Minshuo Chen, Yizhou Wang, Tianyi Liu, Zhuoran Yang Yang, Xingguo Li, Zhaoran Wang, and Tuo Zhao. On computation and generalization of generative adversarial imitation learning. In *International Conference on Learning Representations*, 2020.

Xiaocong Chen, Lina Yao, Aixin Sun, Xianzhi Wang, Xiwei Xu, and Liming Zhu. Generative inverse deep reinforcement learning for online recommendation. In *ACM International Conference on Information & Knowledge Management*, pp. 201–210, 2021.

Wenqiang Chi, Giulio Dagnino, Trevor MY Kwok, Anh Nguyen, Dennis Kundrat, Mohamed EMK Abdelaziz, Celia Riga, Colin Bicknell, and Guang-Zhong Yang. Collaborative robot-assisted endovascular catheterization with generative adversarial imitation learning. In *IEEE International Conference on Robotics and Automation*, pp. 2414–2420, 2020.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adverserial inverse reinforcement learning. In *International Conference on Learning Representations*, 2018.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International Conference on Machine Learning*, pp. 1587–1596, 2018.

Seyed Kamyar Seyed Ghasemipour, Richard Zemel, and Shixiang Gu. A divergence minimization perspective on imitation learning methods. In *Conference on Robot Learning*, pp. 1259–1277, 2019.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, pp. 2672—-2680, 2014.

Yang Guan, Shengbo Eben Li, Jingliang Duan, Jie Li, Yangang Ren, Qi Sun, and Bo Cheng. Direct and indirect reinforcement learning. *International Journal of Intelligent Systems*, 36(8):4439–4467, 2021a.

Ziwei Guan, Tengyu Xu, and Yingbin Liang. When will generative adversarial imitation learning algorithms attain global convergence? In *International Conference on Artificial Intelligence and Statistics*, pp. 1117–1125, 2021b.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, volume 29, pp. 4565–4573, 2016.

Mohamed Khalil Jabri. Robot manipulation learning using generative adversarial imitation learning. In *International Joint Conference on Artificial Intelligence*, pp. 4893–4894, 2021.

Liyiming Ke, Sanjiban Choudhury, Matt Barnes, Wen Sun, Gilwoo Lee, and Siddhartha Srinivasa. Imitation learning as f-divergence minimization. In *International Workshop on the Algorithmic Foundations of Robotics*, pp. 313–329, 2020.

Ilya Kostrikov, Kumar Krishna Agrawal, Debidatta Dwibedi, Sergey Levine, and Jonathan Tompson. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. In *International Conference on Learning Representations*, 2019.

Guy Lever and Ronnie Stafford. Modelling policies in mdps in reproducing kernel hilbert space. In *Artificial intelligence and statistics*, pp. 590–598, 2015.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

Charles A Micchelli and Massimiliano Pontil. On learning vector-valued functions. *Neural Computation*, 17(1):177–204, 2005.

Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning*, pp. 663–670, 2000.

Dirk Ormoneit and Śaunak Sen. Kernel-based reinforcement learning. *Machine learning*, 49(2): 161–178, 2002.

Ling Pan, Qingpeng Cai, and Longbo Huang. Softmax deep double deterministic policy gradients. In *Advances in Neural Information Processing Systems*, volume 33, pp. 11767–11777, 2020.

Santiago Paternain, Juan Andrés Bazerque, Austin Small, and Alejandro Ribeiro. Stochastic policy gradient ascent in reproducing kernel hilbert spaces. *IEEE Transactions on Automatic Control*, 66(8):3429–3444, 2020.

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

Martin L Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

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

Jing-Cheng Shi, Yang Yu, Qing Da, Shi-Yong Chen, and An-Xiang Zeng. Virtual-taobao: Virtualizing real-world online retail environment for reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 4902–4909, 2019.

Richard S Sutton and Andrew G Barto. *Reinforcement Learning: an Introduction*. Cambridge, MA, USA: MIT Press, 2018.

Umar Syed and Robert E Schapire. A game-theoretic approach to apprenticeship learning. In *Advances in Neural Information Processing Systems*, volume 20, pp. 1449–1456, 2007.

Lei Tai, Jingwei Zhang, Ming Liu, and Wolfram Burgard. Socially compliant navigation through raw depth inputs with generative adversarial imitation learning. In *IEEE International Conference on Robotics and Automation*, pp. 1111–1117, 2018.

Yawei Wang and Xiu Li. Reward function shape exploration in adversarial imitation learning: an empirical study. In *IEEE International Conference on Artificial Intelligence and Computer Applications*, pp. 52–57, 2021.

Tian Xu, Ziniu Li, and Yang Yu. Error bounds of imitating policies and environments. In *Advances in Neural Information Processing Systems*, volume 33, pp. 15737–15749, 2020.

Yirui Zhou, Yangchun Zhang, Xiaowei Liu, Wanying Wang, Zhengping Che, Zhiyuan Xu, Jian Tang, and Yaxin Peng. Generalization and computation for policy classes of generative adversarial imitation learning. In *International Conference on Parallel Problem Solving from Nature*, pp. 385–399, 2022.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 8, pp. 1433–1438, 2008.

Guoyu Zuo, Kexin Chen, Jiahao Lu, and Xiangsheng Huang. Deterministic generative adversarial imitation learning. *Neurocomputing*, 388:60–69, 2020.

A Appendix

A.1 Proof of Proposition 1

Proposition 1 *In the MDP $\mathcal{M}$, under the optimal discriminator, PLR of $(s_t, a_{t\to t+1})$ satisfies*

$$r(s_t, a_{t\to t+1}) = -\log(1 - D(s_t, a_{t\to t+1})) \to 0 \quad as \quad |\mathcal{S}| \to \infty,$$

where

$$D(s_t, a_{t\to t+1}) = \begin{cases} \frac{48g}{11g^2+72g-16}, & 0 \le t < \frac{3g}{4}, \\ \frac{16g}{11g^2+40g-16}, & t = \frac{3g}{4}, \\ 0, & \text{otherwise.} \end{cases}$$

Proof. The length of the expert trajectory is

$$\begin{aligned} N &= 2\left(\frac{g}{4} + (\frac{g}{4}+1) + \cdots + (\frac{g}{4}+\frac{g}{4}-1)\right) + 2(\frac{g}{2}+1)\frac{g}{2} - 1 + \frac{3g}{4} \\ &= \frac{11g^2+24g-16}{16}. \end{aligned}$$

Denote the expert policy and the deterministic policy as π_{E} and h, respectively. For any $(s_t, a_{t\to t+1})$, $0 \le t < 3g/4$ in the trajectory of the deterministic policy, the optimal discriminator is (Kostrikov et al., 2019):

$$\begin{aligned} D(s_t, a_{t\to t+1}) &= \frac{\rho^{\pi_{\mathrm{E}}}(s_t, a_{t\to t+1})}{\rho^{\pi_{\mathrm{E}}}(s_t, a_{t\to t+1}) + \rho^{h}(s_t, a_{t\to t+1})} \\ &= \frac{\frac{3}{N}}{\frac{3}{N}+\frac{1}{g}} \\ &= \frac{48g}{11g^2+72g-16}. \end{aligned}$$

For $t = 3g/4$, we have

$$D(s_{3g/4}, a_{3g/4\to 3g/4+1}) = \frac{16g}{11g^2+40g-16}.$$

While for any $t > 3g/4$, the value $D(s_t, a_{t\to t+1})$ is zero since it never appears in the expert demonstrations. To sum up, the discriminator achieves its optimal

$$r(s_t, a_{t\to t+1}) = -\log(1 - D((s_t, a_{t\to t+1}))) \to 0 \quad as \quad |\mathcal{S}| \to \infty.$$

□

A.2 Proof of Theorem 1

Theorem 1 (Probability of Exploding Gradients) *Let $\pi_h(\cdot|s)$ be the Gaussian stochastic policy with mean $h(s)$ and covariance $\mathbf{\Sigma}$. When $D(s,a)$ achieves its optimum in Eq. (6), the gradient estimator of the policy loss with respect to the policy's parameter h is given by*

$$\hat{\nabla}_h D_{\mathrm{JS}}(\rho^{\pi_h}, \rho^{\pi_{\mathrm{E}}}) = \frac{d^{\pi_h}(s_t)\nabla_h \pi_h(a_t|s_t)}{2d^{\pi_{\mathrm{E}}}(s_t)\pi_{\mathrm{E}}(a_t|s_t)} \log \frac{2d^{\pi_h}(s_t)\pi_h(a_t|s_t)}{d^{\pi_h}(s_t)\pi_h(a_t|s_t) + d^{\pi_{\mathrm{E}}}(s_t)\pi_{\mathrm{E}}(a_t|s_t)},$$

where $\nabla_h \pi_h(a|s) = \pi_h(a|s)\kappa(s,\cdot)\mathbf{\Sigma}^{-1}(a - h(s))$. Then we have

$$\|\hat{\nabla}_h D_{\mathrm{JS}}(\rho^{\pi_h}, \rho^{\pi_{\mathrm{E}}})\|_2 \to \infty$$

with a probability of $\Pr(\|\mathbf{\Sigma}^{-1}(a_t - h(s_t))\|_2 \ge C$ *for any* $C > 0)$.

Proof. Through importance sampling which transfers the learned state-action distribution to the expert demonstration distribution, the JS divergence can be written as

$$\begin{aligned} & D_{\mathrm{JS}}(\rho^{\pi_h}, \rho^{\pi_{\mathrm{E}}}) \\ &= \frac{1}{2} D_{\mathrm{KL}}(\rho^{\pi_h}, \frac{\rho^{\pi_h}+\rho^{\pi_{\mathrm{E}}}}{2}) + \frac{1}{2} D_{\mathrm{KL}}(\rho^{\pi_{\mathrm{E}}}, \frac{\rho^{\pi_h}+\rho^{\pi_{\mathrm{E}}}}{2}) \\ &= \frac{1}{2}\mathbb{E}_{(s,a)\sim\mathcal{D}_{\mathrm{I}}}\left[\log \frac{2\rho^{\pi_h}(s,a)}{\rho^{\pi_h}(s,a)+\rho^{\pi_{\mathrm{E}}}(s,a)}\right] + \frac{1}{2}\mathbb{E}_{(s,a)\sim\mathcal{D}_{\mathrm{E}}}\left[\log \frac{2\rho^{\pi_{\mathrm{E}}}(s,a)}{\rho^{\pi_h}(s,a)+\rho^{\pi_{\mathrm{E}}}(s,a)}\right] \\ &= \frac{1}{2}\mathbb{E}_{(s,a)\sim\mathcal{D}_{\mathrm{E}}}\left[\frac{\rho^{\pi_h}(s,a)}{\rho^{\pi_{\mathrm{E}}}(s,a)} \log \frac{2\rho^{\pi_h}(s,a)}{\rho^{\pi_h}(s,a)+\rho^{\pi_{\mathrm{E}}}(s,a)} + \log \frac{2\rho^{\pi_{\mathrm{E}}}(s,a)}{\rho^{\pi_h}(s,a)+\rho^{\pi_{\mathrm{E}}}(s,a)}\right], \end{aligned}$$

where $\mathcal{D}_{\mathrm{E}}$ and $\mathcal{D}_{\mathrm{I}}$ denote the expert demonstration and the replay buffer of π_h respectively. Then the gradient estimator of the JS divergence with respect to h (Guan et al., 2021a) is

$$\hat{\nabla}_h D_{\mathrm{JS}}(\rho^{\pi_h}, \rho^{\pi_{\mathrm{E}}}) = \frac{d^{\pi_h}(s_t)\nabla_h \pi_h(a_t|s_t)}{2d^{\pi_{\mathrm{E}}}(s_t)\pi_{\mathrm{E}}(a_t|s_t)} \log \frac{2d^{\pi_h}(s_t)\pi_h(a_t|s_t)}{d^{\pi_h}(s_t)\pi_h(a_t|s_t) + d^{\pi_{\mathrm{E}}}(s_t)\pi_{\mathrm{E}}(a_t|s_t)},$$

which is obtained by Eq. (1). By the fact that

$$\begin{aligned}\nabla_h \pi_h(a|s) &= \pi_h(a|s)\nabla_h \log \pi_h(a|s) \\ &= \pi_h(a|s)\kappa(s,\cdot)\mathbf{\Sigma}^{-1}(a - h(s)),\end{aligned}$$

it can be shown that

$$\begin{aligned}&\|\hat{\nabla}_h D_{\mathrm{JS}}(\rho^{\pi_h}, \rho^{\pi_{\mathrm{E}}})\|_2 \\ &= \left\| \frac{d^{\pi_h}(s_t)\pi_h(a_t|s_t)\kappa(s_t,\cdot)\mathbf{\Sigma}^{-1}(a_t - h(s_t))}{2d^{\pi_{\mathrm{E}}}(s_t)\pi_{\mathrm{E}}(a_t|s_t)} \log \frac{2d^{\pi_h}(s_t)\pi_h(a_t|s_t)}{d^{\pi_h}(s_t)\pi_h(a_t|s_t) + d^{\pi_{\mathrm{E}}}(s_t)\pi_{\mathrm{E}}(a_t|s_t)} \right\|_2 .\end{aligned}$$

Then, it follows that $\|\hat{\nabla}_h D_{\mathrm{JS}}(\rho^{\pi_h}, \rho^{\pi_{\mathrm{E}}})\|_2 \to \infty$ with a probability of

$$\Pr(\|\mathbf{\Sigma}^{-1}(a_t - h(s_t))\|_2 \geq C \text{ for any } C > 0).$$

□