Imitation Learning from Purified Demonstrations

Yunke Wang [1] **Minjing Dong** [2] **Yukun Zhao** [1] **Bo Du** [1] **Chang Xu** [3]

Abstract

Imitation learning has emerged as a promising approach for addressing sequential decision-making problems, with the assumption that expert demonstrations are optimal. However, in real-world scenarios, most demonstrations are often imperfect, leading to challenges in the effectiveness of imitation learning. While existing research has focused on optimizing with imperfect demonstrations, the training typically requires a certain proportion of optimal demonstrations to guarantee performance. To tackle these problems, we propose to purify the potential noises in imperfect demonstrations first, and subsequently conduct imitation learning from these purified demonstrations. Motivated by the success of diffusion model, we introduce a two-step purification via diffusion process. In the first step, we apply a forward diffusion process to smooth potential noises in imperfect demonstrations by introducing additional noise. Subsequently, a reverse generative process is utilized to recover the optimal demonstration from the diffused ones. We provide theoretical evidence supporting our approach, demonstrating that the distance between the purified and optimal demonstration can be bounded. Empirical results on MuJoCo and RoboSuite demonstrate the effectiveness of our method from different aspects.

1. Introduction

Reinforcement Learning (RL) (Sutton & Barto, 2018; Kaelbling et al., 1996) has achieved significant success in addressing sequential decision-making problems (Silver et al., 2016; Van Hasselt et al., 2016; Zha et al., 2021). The core tenet of RL lies in the learning of an optimal policy via rewarding an agent's actions during its interaction with the environment. The strategic formulation of reward is essential to the recovery of the best policy. However, the intricacy of reward engineering often poses significant challenges in real-world tasks, leading to the failure of RL algorithms occasionally (Amodei et al., 2016; Dulac-Arnold et al., 2021). In response to these challenges, an alternative approach to policy learning is Imitation Learning (IL) (Abbeel & Ng, 2004; Hussein et al., 2017; Cai et al., 2021), a learning framework that utilizes expert behaviors to guide agent learning. One fundamental IL approach is Behavioral Cloning (BC) (Torabi et al., 2018; Sasaki & Yamashina, 2021), in which the agent observes the action of the expert and learns a policy by directly minimizing the action probability discrepancy via supervised learning. This offline training manner has been proven to suffer from compounding error when the agent executes the policy, leading it to drift to new and dangerous states (Xu et al., 2020; 2021). In contrast, Generative Adversarial Imitation Learning (GAIL) (Ho & Ermon, 2016; Fu et al., 2018; Dadashi et al., 2020; Cai et al., 2023; Wang et al., 2023b) has revealed that imitation learning can be framed as a problem of matching state-action occupancy measures, resulting in a more accurate policy. Based on the framework of Generative Adversarial Nets (GAN) (Goodfellow et al., 2014), the discriminator in GAIL is introduced to distinguish demonstrations from expert policy and agent policy, yet the agent policy tries its best to generate behaviors that cheat the judgment of the discriminator.

Current imitation learning approaches have shown promising results under the premise that expert demonstrations exhibit high-quality performance. Nonetheless, acquiring optimal demonstrations can often be costly in practical real-world applications. In many cases, the demonstrations available are not optimal, leading to the problem of imperfect demonstrations in imitation learning. Under such a situation, imitation learning algorithms are prone to failure when expert demonstrations are characterized by noise. Hence, the question of how to effectively learn an optimal policy from imperfect demonstrations becomes central to bridging the application gap of imitation learning from simulated environments to real-world tasks. This issue is critical for enhancing the adaptability and effectiveness of imitation

[1]School of Computer Science, National Engineering Research Center for Multimedia Software, Institute of Artificial Intelligence and Wuhan Institute of Data Intelligence, Wuhan University, China. [2]Department of Computer Science, City University of Hong Kong, China. [3]School of Computer Science, Faculty of Engineering, The University of Sydney, Australia. Correspondence to: Bo Du <dubo@whu.edu.cn>, Chang Xu <c.xu@sydney.edu.au>.

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

learning methods in various practical scenarios.

Confidence-based methods have been shown to be effective when addressing the issue of imperfect demonstrations in imitation learning. In WGAIL (Wang et al., 2021b) and SAIL (Wang et al., 2021a), confidence estimation for each demonstration is tied to the discriminator during the adversarial training phase. BCND (Sasaki & Yamashina, 2021) proposes that the agent's policy itself can estimate confidence. Rather than predicting the confidence directly through either policy or discriminator, CAIL (Zhang et al., 2021) considers confidence as a learnable parameter and jointly optimizes it with imitation learning methods. The key of confidence-based methods lies in how to derive proper confidence for each expert demonstration. However, most existing methods integrate confidence estimation into the training of IL (Beliaev et al., 2022), leading to a bi-level optimization problem. This bi-level optimization can become unstable and hard to converge during the training of imitation learning, resulting in a breakdown of the confidence estimation. Furthermore, the aforementioned methods are specifically designed to be compatible with either BC or GAIL. This specificity limits their flexibility and restricts their application with other methods.

Rather than integrating the handling of imperfect demonstrations into IL training, we propose an approach where the purification of imperfect demonstrations is performed first. Subsequently, imitation learning is carried out with purified demonstrations. Based on this idea, we introduce Diffusion Purified Imitation Learning (DP-IL), which utilizes the forward and reverse diffusion processes to recover the optimal demonstrations from imperfect ones. By incorporating the diffusion process into denoising imperfect demonstrations, we provide theoretical analysis on the effectiveness of noise elimination during the forward diffusion process as well as the reduction of the gap between the optimal and purified distribution. The distance between the purified and optimal demonstration can also be bounded. We show that the purified demonstrations can be used in both online and offline imitation learning methods. Experimental results in MuJoCo (Todorov et al., 2012) and RoboSuite (Zhu et al., 2020) demonstrate the effectiveness of our method from different aspects.

2. Related Work

2.1. Imitation Learning from Imperfect Demonstrations

When dealing with imperfect demonstrations, confidence-based IL methods estimate the weight for imperfect demonstrations to address their importance to agent learning. 2IWIL (Wu et al., 2019) and IC-GAIL (Wu et al., 2019) first investigate the effectiveness of weighting schemes in imitation learning with imperfect demonstrations. However, these approaches relied on manually labeled confidence, which is challenging to obtain in practical scenarios. To relax this requirement, subsequent works have proposed alternative methods. For instance, DWBC (Xu et al., 2022) and DICE-based methods (Kim et al., 2021; Chang et al., 2021; Ma et al., 2022; Yu et al., 2023; Li et al., 2024) leverage a small fraction of known optimal demonstrations to infer the weights for the remaining supplementary demonstrations. CAIL (Zhang et al., 2021) utilizes partially ranked trajectories to guide confidence estimation. Recent works also focus on estimating confidence without exposing too much prior information. WGAIL (Wang et al., 2021b) and SAIL (Wang et al., 2021a) connect confidence estimation to the discriminator during training. BCND (Sasaki & Yamashina, 2021) use the policy network to indicate confidence. However, these approaches have two primary limitations. First, the weight estimation and imitation learning build up a bi-level optimization problem, which can be challenging to converge. Secondly, most methods rely on having a certain proportion of optimal demonstrations, which may not be feasible in practical settings.

Preference-based methods (Christiano et al., 2017; Ibarz et al., 2018) have also proven effective for policy learning from imperfect demonstrations. Using human preference can avoid complex reward engineering, thus making policy learning more practical. T-REX (Brown et al., 2019) focuses on extrapolating a reward function based on ranked trajectories. By effectively capturing the rankings, the learned reward function provides valuable feedback to guide the agent's learning process. T-REX only requires precise rankings of trajectories, yet does not set constraints on data quality. As a result, T-REX can achieve satisfactory performance even in scenarios where optimal trajectories are unavailable. D-REX (Brown et al., 2020) introduces relaxation to the ranking constraint of T-REX. Initially, it learns a pre-trained policy through behavioral cloning and subsequently generates ranked trajectories by injecting varying levels of noise into the actions. D-REX uses the same way to learn the reward as T-REX with ranked trajectories. To address potential ranking errors, SSRR (Chen et al., 2021) proposes a novel reward function structure. By mitigating the adverse effects arising from ranking inaccuracies, SSRR enhances the reliability of the learning process.

2.2. Diffusion Model in Imitation Learning

Diffusion models have demonstrated significant potential in generative tasks. Some works have explored the use of diffusion models either to directly model the policy network or to serve as the discriminator within Adversarial Imitation Learning (AIL) frameworks. For example, policy network in (Chi et al., 2023; Pearce et al., 2022; Reuss et al., 2023) is defined to be a condition diffusion model that refines a noise to the action based on the given state. DiffAIL (Wang

et al., 2024) models the state-action pairs as unconditional diffusion models and uses diffusion loss as part of the discriminator's learning objective. Despite achieving SOTA performance in some benchmark settings, they can not solve imperfect demonstration issues in imitation learning. In (Wang et al., 2023a), the diffusion model is trained with expert demonstrations and used to enhance the generalization of policy trained by BC. Despite both (Wang et al., 2023a) and our method trains a diffusion model on expert demonstrations first, the diffusion model is used to achieve different tasks.

3. Preliminary

Before diving into our method, we briefly review the definition of the Markov Decision Process (MDP) and Imitation Learning with Distribution Matching.

3.1. Markov Decision Process (MDP)

MDP is popular for formulating reinforcement learning (RL) (Puterman, 1994) and imitation learning (IL) problems. An MDP normally consists six basic elements $M = (\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \mu_0)$, where $\mathcal{S}$ is a set of states, $\mathcal{A}$ is a set of actions, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the stochastic transition probability from current state s to the next state s', $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the obtained reward of agent when taking action a in a certain state s, $\gamma \in [0, 1]$ is the discounted rate and $\mu_0 : \mathcal{S} \to [0, 1]$ denotes the initial state distribution.

Definition 1. For any policy $\pi(a|s) : \mathcal{S} \to \mathcal{A}$, there is an one-to-one correspondence between π and its occupancy measure $\rho_\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$, which is formulated as

$$\rho_\pi(s, a) = (1 - \gamma)\pi(a|s) \sum_{t=0}^{\infty} \gamma^t Pr(s_t = s|\pi), \quad (1)$$

where $Pr(s_t = s|\pi)$ denotes the probability density of state s at timestep t following policy π.

3.2. Imitation Learning via Distribution Matching

The field of Imitation Learning (IL) is concerned with optimizing an agent's behavior in a given environment by utilizing expert demonstrations. Given expert demonstrations $\mathcal{D}_e$ sampled from the expert policy π_e, imitation learning methods aim to let the agent policy π_θ replicate the expert behavior. Distribution Matching (DM) approaches attempt to match the agent's state-action distribution ρ_{π_θ} with that of the expert's ρ_{π_e} by minimizing the f-divergence (Nowozin et al., 2016; Ke et al., 2019),

$$\theta^* = \arg\min_\theta D_f(\rho_{\pi_e}(s, a), \rho_{\pi_\theta}(s, a)) \quad (2)$$

where $f : \mathbb{R}^+ \to \mathbb{R}$ is a convex, lower semi-continuous function and satisfies $f(1) = 0$. Different choices of f-divergence can recover different imitation learning methods. For example, using KL divergence, Jensen-Shannon divergence can recover Behavior Cloning (BC) (Sasaki & Yamashina, 2021), Generative Adversarial Imitation Learning (GAIL) (Ho & Ermon, 2016), respectively.

4. Methodology

Imitation learning achieves promising results in benchmark tasks with two non-trivial assumptions: (i) Expert demonstrations are sampled from an optimal policy, and (ii) Expert demonstrations are enough to encompass the expert distribution. While these two assumptions may not always hold in practice, our setting falls into the problem of imitation learning with imperfect demonstrations, where we have access to a small fraction of optimal demonstrations and a large fraction of supplementary sub-optimal demonstrations. To tackle this problem, we propose a two-step method named Diffusion Purified Imitation Learning (DP-IL), in which sub-optimal demonstrations are purified via a combination of forward and reverse diffusion process. Furthermore, we provide theoretical results on the optimal reverse point that could achieve best performance.

4.1. General Objective

We begin by formalizing the problem of incorporating supplementary sub-optimal demonstrations into imitation learning. For simplicity, we denote the optimal expert distribution as ρ_{π_o} and sub-optimal expert distribution as ρ_{π_s}. In this setting, we have access to a limited number of optimal demonstrations $\mathcal{D}_o = \{s_i, a_i\}_{i=1}^{n_o} \sim \rho_{\pi_o}(s, a)$, and a large number of supplementary sub-optimal demonstrations $\mathcal{D}_s = \{s_i, a_i\}_{i=1}^{n_s} \sim \rho_{\pi_s}(s, a)$. Our objective is to leverage both types of demonstrations $\mathcal{D}_e = \mathcal{D}_o \cup \mathcal{D}_s$ to learn a good policy π for the agent. Typically, $\mathcal{D}_o$ alone is often insufficient for effective policy learning (Kim et al., 2021; Xu et al., 2022), and the involvement of $\mathcal{D}_s$ could significantly hurt the optimization of imitation learning algorithms.

To tackle this problem, a potential solution is to purify supplementary sub-optimal demonstrations $\mathcal{D}_s$ under the guidance of optimal demonstrations $\mathcal{D}_o$. Specifically, assuming that $\rho_{\pi_s}(s, a)$ can be purified through a distribution transformation $\mathcal{F}$, the agent policy π_θ can subsequently learn from the purified demonstrations. To summarize, the objective can be formulated as,

$$\min_\theta D_f(\mathcal{F}^*(\rho_{\pi_s}(s, a)), \rho_{\pi_\theta}(s, a)) \quad (3)$$

$$\text{s.t. } \mathcal{F}^* = \arg\min_{\mathcal{F}} \mathcal{H}(\rho_{\pi_o}(s, a), \mathcal{F}(\rho_{\pi_s}(s, a))), \quad (4)$$

where $\mathcal{H}$ denotes the distance measurement between distributions. We assume that sub-optimality within supplementary demonstrations mainly come from the potential perturbations δ during the collections, which form the sub-optimal expert distribution. Thus, taking $\mathcal{F}^*$ as a denoising

function can be a potential solution to tackle the gap between optimal and sub-optimal expert distributions.

Motivated by the success of diffusion models in adversarial purification (Nie et al., 2022), we introduce a Diffusion Purified Imitation Learning (DP-IL) algorithm, which eliminates the potential noises in ρ_{π_s} through a two-step diffusion process. In the diffusion model, the first step involves a forward diffusion process, where small timestep noise is gradually added to the sub-optimal demonstrations. This results in diffused demonstrations that eliminate potential perturbation patterns in the sub-optimal expert distribution ρ_{π_s} while preserving the original semantic information. In the second step, a reverse-time diffusion process is applied to reconstruct the purified demonstrations from the diffused demonstrations.

4.2. Purification via Diffusion Process

We formulate $\mathcal{F}^*$ as a two-step purification as in the diffusion model, which includes forward and reverse diffusion processes. For the diffusion model, we adopt widely-used DDPM (Ho et al., 2020). For simplicity, we use x to represent the state-action pair (s, a) in the following subsections since we regard it as a whole during the purification process.

Training Diffusion Model with Optimal Demos. Since the transformation $\mathcal{F}$ is a combination of forward and reverse diffusion processes, we train a diffusion model ϵ_ϕ on optimal demonstrations x_o from ρ_{π_o} first. To simplify the formulation, we concatenate the state-action pair to construct the latent variable x for the diffusion model $\epsilon_\phi(x_i, i)$. Subsequently, we inject noises ϵ on x_o, where i indicates the number of steps of the Markov procedure in the DDPM , which can be viewed as a variable of the level of noise. The goal of the diffusion model is to reverse the diffusion process (*i.e.*, denoise), yielding the learning objective,

$$\min_{\phi} \mathbb{E}_{x_o,\epsilon,i}[\epsilon - \epsilon_\phi(\sqrt{\bar{\alpha}_i} \cdot x_o + \sqrt{1-\bar{\alpha}_t}\epsilon, t)]^2, \quad (5)$$

where i is sampled from a uniform distribution, ϵ is the Gaussian noise sampled from $\mathcal{N}(0, I)$, α_i is defined to be $\alpha_i = 1 - \beta_i$ and β_i is the noise schedule which monotonically increase with i. $\bar{\alpha}_i$ is defined to be $\bar{\alpha}_i = \prod_{s=1}^{i} \alpha_s$.

Purifying Sub-optimal Demonstrations. After we obtain the trained diffusion model ϵ_ϕ, we purify sub-optimal demonstrations x_s sample from ρ_{π_s} with ϵ_ϕ. Supposing the inverse point $i_r \in \{0, ..., N-1\}$ is the intermediate step that is used in the diffusion purification process, its forward diffusion can be calculated as follows,

$$\hat{x}_{i_r} = \sqrt{\bar{\alpha}_{i_r}} \cdot x_s + \sqrt{1-\bar{\alpha}_{i_r}}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I). \quad (6)$$

Then, the reverse diffusion process is used to denoise $\hat{x}_{i_r}$ to $\hat{x}_o$ from timestep i_r to 0. The reverse diffusion at every step is defined as

$$\hat{x}_{i-1} = \frac{1}{\sqrt{\alpha_i}}\Big(\hat{x}_i - \frac{1-\alpha_i}{\sqrt{1-\bar{\alpha}_i}\epsilon_\phi(\hat{x}_i, t)}\Big) + \sqrt{\beta_i}z, \quad (7)$$

where $z \sim \mathcal{N}(0, I)$. Starting from intermediate step $\hat{x}_{i_r}$, we recover the $\hat{x}_0$ from Eq. 7, which is the purified demonstration corresponding to x_s.

Choice of optimal i_r Intuitively, setting a larger i_r can help to smooth the potential noise within sub-optimal demonstrations in the forward diffusion. When $i_r = N-1$, the purified demonstrations can be seen as samples randomly drawn from the diffusion model ϵ_ϕ. However, training such a diffusion model ϵ_ϕ exclusively with limited optimal demonstrations may result in generated demonstrations lacking diversity and coverage, encountering similar issues as imitation learning solely from $\mathcal{D}_o$. A properly chosen i_r achieves good trade-off between denoising (Eq. 6) and maintaining the structure of sub-optimal demonstrations (Eq. 7), thus can help the agent achieve better performance.

Notice that DDPM is a discretization of VP-SDE (Song et al., 2021b), and the step i in DDPM is a discrete value that varies from 0 to $N-1$. In the theoretical analysis below, we extend DDPM to VP-SDE and scale the timestep to $t \in [0, 1]$. Hence, the problem is to find an optimal t_r^*.

4.3. Theoretical Analysis

In the above subsection, we propose to utilize a two-step purification to tackle supplementary sub-optimal demonstrations in imitation learning. While the choice of optimal t_r^* remains unknown, we further provide a more theoretical analysis on (i) the impact of t_r^* on the purification (Theorem 1 and Theorem 2), and (ii) how different levels of noises affect the choice of t_r^* (Theorem 4).

We first analyze the purification effectiveness with respect to t during the forward diffusion process. Considering the optimal expert distribution ρ_{π_o} and the sub-optimal one ρ_{π_s}, we show that the gap between the diffused distribution $\rho_{\pi_o,t}$ and $\rho_{\pi_s,t}$ becomes smaller with increasing timestep t, which implies that the potential perturbations can be smoothed via the gradually added noises during the forward process.

Theorem 1. *Let $\{x_t\}_{t\in\{0,1\}}$ be samples in the forward diffusion process. If we denote $\rho_{\pi_o,t}(x)$ and $\rho_{\pi_s,t}(x)$ as the respective distributions of x_t when $x_{o,0} \sim \rho_{\pi_o,t=0}(x)$ and $x_{s,0} \sim \rho_{\pi_s,t=0}(x)$, we then have,*

$$\varsigma \leq -\frac{1}{2}\int \rho_{\pi_o,t}(x)\beta_t||\nabla_x \log \rho_{\pi_o,t}(x) - \nabla_x \log \rho_{\pi_s,t}(x)||_2^2 \mathrm{d}x, \quad (8)$$

where $\varsigma = \frac{\partial D_{KL}(\rho_{\pi_o,t}(x)||\rho_{\pi_s,t}(x))}{\partial t}$ denotes the derivative of t to the KL divergence between $\rho_{\pi_o,t}(x)$ and $\rho_{\pi_s,t}(x)$.

Theorem 1 is derived from DiffPure (Nie et al., 2022). The difference is that we start from the forward diffusion step in DDPM and extend discrete steps to continuous steps. The *Fisher* divergence $D_{fisher}(p||q)$ between two data distribution p and q is defined to be $\int p||\nabla \log p - \nabla \log q||_2^2$. Therefore, we have $\varsigma \leq -\frac{1}{2} D_{fisher}(\rho_{\pi_o,t}(x)||\rho_{\pi_s,t}(x)) \leq 0$. The equality is only satisfied when $\rho_{\pi_o,t=1}(x) = \rho_{\pi_s,t=1}(x)$. Notice that $\varsigma \leq 0$ indicates that the KL divergence between $\rho_{\pi_o,t}(x)$ and $\rho_{\pi_s,t}(x)$ monotonically decreases towards $t = 0$ to $t = 1$ during the forward diffusion step. In other words, with a relatively larger t in the forward diffusion process, the divergence between diffused optimal and sub-optimal expert distributions can be minimized.

The monotonic decrease of $D_{KL}(\rho_{\pi_o,t}(x)||\rho_{\pi_s,t}(x))$ towards t implies that the diffused optimal expert distribution and diffused sub-optimal expert distribution become closer as t increases. Suppose there exists an κ such that the difference between the diffused optimal expert distribution and diffused sub-optimal expert distribution becomes negligible when $D_{KL}(\rho_{\pi_o,t}(x)||\rho_{\pi_s,t}(x)) \leq \kappa$. We can then find a minimum $t_r \in [0, 1]$ to achieve this. Starting from $\rho_{\pi_s,t_r}(x)$, we can stochastically recover $\rho_{\pi_o}(x)$ at $t = 0$ through the reverse diffusion process. We further provide theoretical evidence that the distance between the purified demonstration and its optimal version can be bounded via the two-step purification.

Theorem 2. *Supposing the sub-optimal demonstration x_s perturbed by noise δ compared to the optimal demonstration x_o, and $\hat{x}_o$ is a purified demonstration of x_s, the L2 distance between x_o and $\hat{x}_o$ (i.e., $\|x_o - \hat{x}_o\|$) is bounded within the interval $[\max\{0, \lambda\}, \Lambda]$ with a probability of at least $1 - \sigma$, where*

$$\lambda(\zeta, t_r, \delta, \sigma) = \zeta(t_r)\tilde{C}_{gd} - \|\delta\| - \sqrt{e^{2\zeta(t_r)} - 1} C_{\sigma,d}, \tag{9}$$

$$\Lambda(\zeta, t_r, \delta, \sigma) = \zeta(t_r)C_{gd} + \|\delta\| + \sqrt{e^{2\zeta(t_r)} - 1} C_{\sigma,d}, \tag{10}$$

and $\tilde{C}_{gd} \leq 2\|\nabla_x \log \rho_t(x)\| \leq C_{gd}$, $\zeta(t_r) = \int_0^{t_r} \frac{1}{2}\beta_s \mathrm{d}s$, $C_{\sigma,d} = \sqrt{2d(1 + \sqrt{\frac{8\ln(2/\sigma)}{d}})}$, $\sigma \geq 2e^{-d/8}$.

As shown in Theorem 2, the L2 distance between the optimal demonstration x_o and the purified demonstration $\hat{x}_o$ can be bounded by the terms related to the intermediate timestep t_r. According to Theorem 1, t_r should be set to a relatively large value to guarantee that the potential perturbations are removed in diffused distributions, however, t_r cannot be arbitrarily large.

Proposition 3. *$\lambda(\zeta, t_r, \delta, \sigma)$ and $\Lambda(\zeta, t_r, \delta, \sigma)$ are monotonically increasing with respect to t_r.*

Proposition 3 suggests that as t_r increases, the bound of $\|x_o - \hat{x}_o\|$ increases as well. Thus, a relatively small t_r is required to achieve a smaller lower and upper bound to satisfy the minimum L2 distance. Together with the analysis in Theorem 1, there exists a trade-off on t_r between perturbation purification and recovery performance. An optimal timestep t_r^* is expected to meet the requirements that the potential perturbation can be eliminated in diffused distribution while the L2 distance with the related optimal demonstration can also be minimized.

Theorem 4. *There exists a inflection point t_p that satisfies $\lambda''(t_p) = 0$, which makes $\lambda(t_r)$ increases smoothly before t_p while increasing rapidly after t_p. The optimal t_r^* is around the inflection point t_p, or is around solution of the equation*

$$-\frac{(\zeta'(t_r))^2}{\zeta''(t_r)}(e^{2\zeta(t_r)} - 1)^{-3/2} + (e^{2\zeta(t_r)} - 1)^{-1/2} = \frac{\tilde{C}_{gd}}{C_{\sigma,d}}. \tag{11}$$

With a probability of $1 - \sigma$, there is a positive correlation between δ and t_r^*.

As demonstrated in Theorem 4, the lower bound $\lambda(t_r)$ exhibits a smooth increase initially, implying that selecting a larger t_r before the inflection point t_p is unlikely to significantly impact $\lambda(t_r)$. Combined with the analysis in Theorem 1 that shows larger t_r would better smooth the perturbation, we make a conclusion that the optimal t_r^* should be in the vicinity of the inflection point t_p.

Furthermore, Theorem 4 establishes a positive correlation between $\|\delta\|$ and t_r^*, suggesting that for more substantial perturbations within sub-optimal demonstrations, the optimal t_r^* tends to be relatively larger. Empirical results from the experiments also align with and support this observation.

We further provide sensitive analysis of σ in Theorem 5. The proofs of all theorems are provided in the appendix.

5. Experiments

In this section, we conduct extensive experiments to verify the effectiveness of DP-IL in MuJoCo (Todorov et al., 2012) and Robosuite (Zhu et al., 2020) with different compared methods. The experimental results demonstrate the advantage of DP-IL from different aspects.

Benchmarking We first conduct experiments on MuJoCo benchmarks in OpenAI Gym (Brockman et al., 2016). Four popular tasks (*i.e.*, Ant-v2, HalfCheetah-v2, Walker2d-v2 and Hopper-v2) are used to evaluate DP-IL. The evaluated performance in MuJoCo benchmark can be measured by the average cumulative ground-truth rewards (*i.e.*, the higher the better). We repeat experiments for 5 trials with different random seeds for common practice. Additionally, to verify the robustness of DP-IL with real-world human operation demonstrations, we also conduct experiments on a robot control task in Robosuite platform.

Table 1. Performance of DP-BC and compared offline imitation learning methods in four MuJoCo tasks with different sub-optimal demonstrations. 'L1', 'L2' and 'L3' denotes different levels of perturbations within sub-optimal demonstrations. We also provide the best inverse step i_r^* of diffusion purification, and we have $i_r = Nt_r$.

Task	Demos	BC-opt	BC-all	BCND	DWBC	DemoDICE	DP-BC	i_r^*
Ant-v2	D1-L1	29±18	-22±33	28±28	-156±95	-72±8	**261**±54	100
	D1-L2	-	292±88	199±37	132±39	-3±14	**803**±114	50
	D1-L3	-	1500±298	1951±311	1669±244	94±4	**2547**±118	10
	D2-L1	-	1089±237	1539±150	**1545**±128	262±152	1402±151	30
	D2-L2	-	918±232	1514±232	1725±214	480±139	**1982**±111	10
	D2-L3	-	1356±137	2232±324	2484±29	2127±260	**3414**±40	5
HalfCheetah-v2	D1-L1	-254±103	1262±202	1267±202	618±204	22±106	**1365**±147	10
	D1-L2	-	1264±255	1862±119	1069±264	622±159	**2440**±274	10
	D1-L3	-	859±233	834±249	694±190	768±233	**4042**±80	10
	D2-L1	-	1314±194	1498±246	671±178	-64±105	**1530**±39	30
	D2-L2	-	2789±14	2241±245	2676±20	1771±158	**2714**±15	30
	D2-L3	-	1780±296	1990±408	4722±37	4508±59	**4748**±40	10
Walker2d-v2	D1-L1	-4±0	21±2	102±10	1039±164	365±39	**1697**±219	10
	D1-L2	-	1551±87	1635±319	1617±152	1560±291	**1722**±297	10
	D1-L3	-	815±277	192±37	2322±369	583±108	**3020**±466	5
	D2-L1	-	500±245	686±301	1656±186	1749±259	**2208**±160	30
	D2-L2	-	578±273	1018±347	2114±21	1818±342	**3162**±20	5
	D2-L3	-	702±244	2574±358	2637±134	1765±528	**3076**±205	3
Hopper-v2	D1-L1	320±0	743±83	590±10	761±12	995±119	**1000**±42	10
	D1-L2	-	2044±157	2119±206	826±22	2059±4	**3145**±11	5
	D1-L3	-	3090±37	2883±26	1140±27	**3207**±7	1825±137	1
	D2-L1	-	2191±118	2165±181	761±122	2336±3	**2408**±113	10
	D2-L2	-	2266±3	2256±2	826±2	2265±8	**2323**±2	10
	D2-L3	-	2712±6	3093±10	1140±27	**3306**±2	2265±171	5

Source of Demonstrations In our setting, we have access to limited number of optimal demonstrations $\mathcal{D}_o$ and a lot supplementary sub-optimal demonstrations $\mathcal{D}_s$. To form $\mathcal{D}_o$, an optimal policy $\pi_o(a|s)$ trained by TRPO (Schulman et al., 2015) to sample optimal demonstrations. As for $\mathcal{D}_s$, we use two different kinds of ways to generate sub-optimal behaviors following existing works (Tangkaratt et al., 2020; Wu et al., 2019):

- D1: We add Gaussian noise to optimal action distribution a^* of π_o to form π_s. The action of π_s is modeled as $a \sim \mathcal{N}(a^*, \delta)$ and we choose $\delta = [0.6, 0.4, 0.25]$ to form $\mathcal{D}_s$ with varying quality.
- D2: We save checkpoints during the RL training as the sub-optimal policy π_s.

In addition, since we use optimal demonstrations $\mathcal{D}_o$ to train the diffusion model, we include $\mathcal{D}_o$ into the training of compared methods for a fair comparison.

Compared Methods Purified demonstrations are applied to both the offline imitation learning method (*e.g.*, BC) and the online imitation learning method (*e.g.*, GAIL), to verify the generalization of DP-IL. In the case of offline imitation learning, we compare our method against several state-of-the-art offline methods, including BCND (Sasaki & Yamashina, 2021), DWBC (Xu et al., 2022) and DemoDICE (Kim et al., 2021). For online imitation learning, we compare our method with several confidence-based methods, including 2IWIL/IC-GAIL (Wu et al., 2019) and WGAIL (Wang et al., 2021b). Further details (*e.g.*, implementation of DP-IL and compared methods, data quality, and more results) can be found in the appendix.

5.1. Evaluations on MuJoCo

We evaluated the effectiveness of DP-BC under the offline IL setting, and results are presented in Table 1. As have introduced in the experimental setup, we use two different ways to generate sub-optimal demonstrations (*i.e.*, D1 and D2). BC-opt means we only use $\mathcal{D}_o$ to conduct BC training, while BC-all means we use both $\mathcal{D}_o$ and $\mathcal{D}_s$ to train the policy network. Our findings align with those results reported in (Kim et al., 2021), showing that using only $\mathcal{D}_o$ can sometimes yield worse performance compared to BC-all. This can be attributed to the insufficient coverage of the entire optimal demonstration space by a limited number of optimal demonstrations. BC-all benefits from a larger training dataset and can outperform BC-opt in certain cases. However, the challenge of sub-optimal demonstrations still hampers BC's

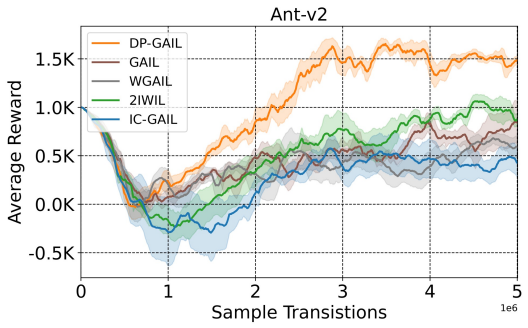

Figure 1. The training curve of DP-GAIL and other online imitation learning methods with D1-L1 demonstrations. The x-axis is the number of interactions with the environment and the shaded area indicates the standard error.

ability to achieve optimal performance. BCND outperforms the baseline at some cases due to its self-weighting strategy. While this strategy is unstable when dealing with a high ratio of sub-optimal demonstrations, we can observe that it sometimes collapses. DWBC and DemoDICE demonstrate great performance with D2 demonstrations with relatively smaller perturbations (*e.g.*, L3), however, we find it is more likely to fail when dealing with sub-optimal demonstrations with strong perturbations. In contrast, our method purifies sub-optimal demonstrations before conducting BC, which consistently improves the subsequent performance of behavior cloning across all different types of sub-optimal demonstrations compared to other methods. This demonstrates the efficacy of our approach in enhancing the performance of BC. Furthermore, the optimal reverse point i_r^* is provided in the table, and the results of DP-BC is related to this i_r^*. As 'L1', 'L2' and 'L3' represent sub-optimal demonstrations with progressively less perturbation, Theorem 4 predicts a corresponding decrease in the optimal reverse point, i_r^*. This pattern is generally observed, illustrating the theorem's validity. The impact of optimal i_r^* will be discussed in the following ablation study in Sec 5.2.

Besides offline imitation learning, we also utilize diffusion-purified demonstrations for GAIL training and compare DP-GAIL with other state-of-the-art online imitation learning methods that also address the imperfect demonstration issue. The training curves of DP-GAIL and other compared methods are depicted in Figure 1. The curve unequivocally demonstrates the superiority of DP-GAIL over other compared methods. This shows the efficacy of noise purification in improving the performance of online imitation learning methods.

5.2. Ablation Study

Different Types of Noises In D1, sub-optimal demonstrations are formed by adding Gaussian noise into the optimal action. It would be interesting to further explore more noises.

Table 2. The performance of DP-BC with demonstrations in different noises.

Task	Demons	BC	BCND	DP-BC
Ant	Uni-L1	209±138	66±17	**343±16**
	Uni-L2	396±42	333±60	**480±15**
	Uni-L3	704±97	1472±229	**1925±227**
HalfCheetah	Uni-L1	704±71	692±37	**733±8**
	Uni-L2	922±207	705±179	**1248±54**
	Uni-L3	1805±15	1933±21	**2268±11**
Ant	S&P-L1	209±138	686±45	**744±17**
	S&P-L2	396±42	477±62	**508±12**
	S&P-L3	704±97	**755±143**	689±37
HalfCheetah	S&P-L1	2143±10	2042±113	**2731±18**
	S&P-L2	1964±195	1873±270	**3550±18**
	S&P-L3	579±280	1160±295	**2820±383**

Table 3. The performance of DP-BC on Ant-v2 task when defining different $\mathcal{F}^*$ in Eq. 3. 'G-Filter', 'M-Filter' and 'Med-Filter' denote using Gaussian, Mean and Median Filter as $\mathcal{F}^*$ to denoise sub-optimal demonstrations.

Demons	BC	DP-BC	G-Filter	M-Filter	Med-Filter
D1-L1	-22	**261**	242	182	138
D1-L2	292	**803**	729	736	437
D1-L3	1500	**2547**	771	1786	455

We consider adding uniform noise and salt-and-pepper noise. Similar to D1, we set $\delta = [0.25, 0.4, 0.6]$ to form L1 to L3 demonstrations. From the Table 2, we can observe that DP-BC performs best in most cases. This suggests the robustness of DP-BC when dealing with different kinds of noises. Intuitively, the noises added to the current data are overwhelmed by the accumulating Gaussian noise during the diffusion forward process, ultimately making the noise less prominent.

Compared with Other Purified Methods The transformation $\mathcal{F}(\cdot)$ is defined as a combination of the forward and reverse diffusion processes to denoise sub-optimal demonstrations. Apart from the diffusion model, there are various denoising methods that can be employed as the transformation $\mathcal{F}^*$. Since filters are commonly used to smooth noisy data, we utilize three different filters (mean, median, and Gaussian) as $\mathcal{F}^*$ to denoise the imperfect demonstrations. The results are presented in Table 3. It is evident from the table that our method outperforms other filter-based methods in all types of sub-optimal demonstrations (*e.g.*, D1-L1, D1-L2 and D1-L3). Among the filter-based methods, the 'Mean Filter' achieves the best performance. Another notable finding is that when the noise level is high in sub-optimal demonstrations (*e.g.*, L1), filter-based methods tend to maintain a greater advantage over the baseline.

Multiple Demonstrators In addition to collecting sub-optimal demonstrations from a single demonstrator, we also investigate the performance of DP-BC when confronted with sub-optimal demonstrations that are sampled from multiple

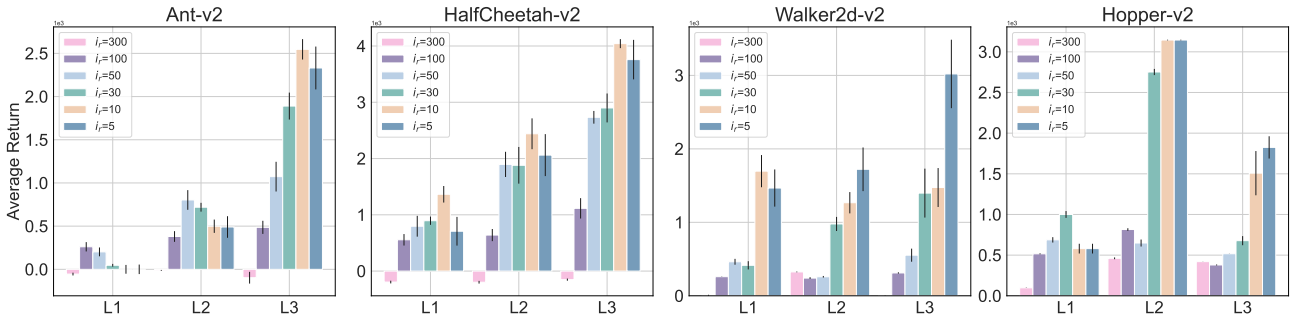

Figure 2. Impact of diffusion time t_r with demonstrations of different optimality.

Table 4. The performance of DP-BC with mixed demonstrations (*i.e.*, D1-L1, D1-L2, and D1-L3).

Task	BC	BCND	DWBC	DemoDICE	DP-BC
Ant	193	411	339	-53	601
HalfCheetah	1255	921	265	344	3146
Walker2d	169	441	290	488	1390
Hopper	587	564	541	605	655

demonstrators. The results are presented in Table 4, where the mixed demonstrations denote a combination of L1, L2, and L3 demonstrations. As shown in the table, DP-BC with demonstrations sampled from a mixture of dataset also consistently outperforms the baseline and other compared methods by a substantial margin.

Impact of optimal i_r As discussed in Section 4.3, i_r is an important hyper-parameter to achieve the trade-off between smoothing the noise and keeping semantic information. To investigate how the choice of i_r affects the effectiveness of noise purification, we conducted experiments on demonstrations with different levels of optimality. Our empirical results confirmed the theoretical findings of Theorem 1, suggesting that demonstrations with less optimality require a relatively larger timestep t_r to smooth the noise. For example, in Ant-v2 task, the optimal t_r is 100 for D1-L1 demonstrations while the optimal t_r is 10 for D1-L3 demonstrations. As the quality of demonstrations improved (e.g., from L1 to L3), we observed a gradual decrease in the optimal value of i_r. This could lead to a smaller value bound since $\zeta(i_r)$ exhibits a monotonic increase with respect to t_r. Hence, diffusion purified imitation learning should perform better when using a smaller i_r under such a case, which is consistent with the results in Figure 2.

5.3. Evaluations on RoboSuite Platform

We also evaluate the robustness of DP-BC on the RoboSuite platform (Zhu et al., 2020) with real-world demonstrations. We consider a reaching task within the "Nut Assembly" in Saywer. In this task, the goal for the robot is to move a nut close to the peg. It earns a higher reward if the nut is closer to the peg. During the reaching phase, the Sawyer robot's arm operates with the gripper in a fully open state, not actively attempting to grasp any objects. Its role is merely to move the nut across the plane. As such, the grasp command is not engaged and prevents the robot's arm from inserting the nut into the peg. Hence, an episode in the reaching task completes only when the timestep matches the horizon length.

Table 5. Successful rate of DP-BC in RoboSuite platform with human demonstrations.

Method	BC	BCND	DWBC	DP-BC (Ours)
Success Rate	0.76	0.56	0.82	**0.86**
i_r	1	3	5	10
DP-BC	0.78	**0.86**	0.74	0.60

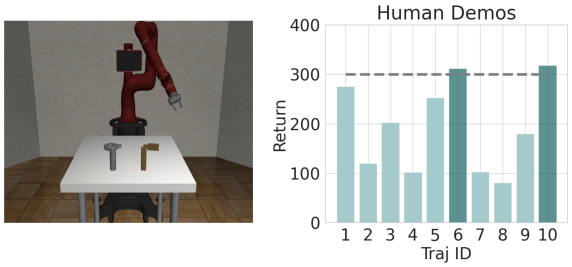

Figure 3. Visualization of SaywerNutAssembly task in RoboSuite platform and the quality of human demonstrations.

We use real-world demonstrations by human operators from RoboTurk (Mandlekar et al., 2018). The demonstrations contain 10 trajectories with approaching length (around 500 state-action pairs). Based on the accumulative reward of trajectories in Figure 3, we regard two trajectories that have the larger reward (*e.g.*, >300) as optimal demonstrations and the remaining trajectories are regarded as supplementary sub-optimal demonstrations. In this reaching task, we report sucess rate rather than reward to assess learned policy. We set the termination condition of the episode to be when the distance between the nut and the peg is less than 0.1. Based on this criterion, we can provide the success rates of different methods, which are calculated over 50 sampled trajectories. The results are shown in Table 5, and we can observe that the proposed method has higher success rate

than compared methods like BCND and DWBC. We also conduct an ablation study on the timestep i_r in this task and we find that setting $i_r = 3$ leads to the best performance.

The results presented in Table 8 also support our theorem about there exists a trade-off in finding an optimal i_r. To further evaluate the relationship between i_r^* and different levels of sub-optimal demonstrations, we divided the sub-optimal demonstrations into two groups and evaluate their performance in the appendix (Table 8).

6. Conclusion

In this paper, we propose to tackle the imperfect demonstrations issue by conducting a two-step purification process to eliminate the potential noises based on the diffusion process. The distance gap between optimal and sub-optimal expert distributions can be minimized after forward diffusion. The purified demonstrations can then be recovered from diffused one via reverse diffusion. Additionally, we provide sufficient theoretical analysis to indicate the impact of the reverse point on the purification. The proposed method can be easily adopted in existing imitation learning frameworks, such as GAIL and BC, to alleviate the effect of sub-optimal expert demonstrations. We conduct extensive experiments on MuJoCo and RoboSuite with different types of sub-optimal demonstrations to evaluate the effectiveness of diffusion purification. The comparison results demonstrate the superiority of DP-IL over other baselines.

Impact Statement

Our work enables agents to learn from a broader range of imperfect data. There are many potential societal consequences of our work, and the safety of learning from such purified data remains an area worthy of further investigation.

Acknowledgements

This work was supported by the National Key Research and Development Program of China 2023YFC2705700, National Natural Science Foundation of China under Grants 62225113, the Innovative Research Group Project of Hubei Province under Grants 2024AFA017, the Australian Research Council under Projects DP210101859 and FT230100549, and the CityU APRC Project No. 9610680.

References

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

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and Mané, D. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Beliaev, M., Shih, A., Ermon, S., Sadigh, D., and Pedarsani, R. Imitation learning by estimating expertise of demonstrators. In *International Conference on Machine Learning*, pp. 1732–1748. PMLR, 2022.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym, 2016.

Brown, D. S., Goo, W., Nagarajan, P., and Niekum, S. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. *arXiv preprint arXiv:1904.06387*, 2019.

Brown, D. S., Goo, W., and Niekum, S. Better-than-demonstrator imitation learning via automatically-ranked demonstrations. In *Conference on Robot Learning*, pp. 330–359, 2020.

Cai, X.-Q., Ding, Y.-X., Jiang, Y., and Zhou, Z.-H. Imitation learning from pixel-level demonstrations by hashreward. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 279–287, 2021.

Cai, X.-Q., Ding, Y.-X., Chen, Z., Jiang, Y., Sugiyama, M., and Zhou, Z.-H. Seeing differently, acting similarly: Heterogeneously observable imitation learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Chang, J., Uehara, M., Sreenivas, D., Kidambi, R., and Sun, W. Mitigating covariate shift in imitation learning via offline data with partial coverage. *Advances in Neural Information Processing Systems*, 34:965–979, 2021.

Chen, L., Paleja, R., and Gombolay, M. Learning from suboptimal demonstration via self-supervised reward regression. In *Conference on Robot Learning*, pp. 1262–1277. PMLR, 2021.

Chi, C., Feng, S., Du, Y., Xu, Z., Cousineau, E., Burchfiel, B., and Song, S. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, pp. 4299–4307, 2017.

Dadashi, R., Hussenot, L., Geist, M., and Pietquin, O. Primal wasserstein imitation learning. In *International Conference on Learning Representations*, 2020.

Dulac-Arnold, G., Levine, N., Mankowitz, D. J., Li, J., Paduraru, C., Gowal, S., and Hester, T. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning*, 110(9):2419–2468, 2021.

Fu, J., Luo, K., and Levine, S. Learning robust rewards with adverserial inverse reinforcement learning. In *International Conference on Learning Representations*, 2018.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

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

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

Hussein, A., Gaber, M. M., Elyan, E., and Jayne, C. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

Ibarz, B., Leike, J., Pohlen, T., Irving, G., Legg, S., and Amodei, D. Reward learning from human preferences and demonstrations in atari. *Advances in neural information processing systems*, 31, 2018.

Kaelbling, L. P., Littman, M. L., and Moore, A. W. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

Ke, L., Barnes, M., Sun, W., Lee, G., Choudhury, S., and Srinivasa, S. Imitation learning as f-divergence minimization. *arXiv preprint arXiv:1905.12888*, 2019.

Kim, G.-H., Seo, S., Lee, J., Jeon, W., Hwang, H., Yang, H., and Kim, K.-E. Demodice: Offline imitation learning with supplementary imperfect demonstrations. In *International Conference on Learning Representations*, 2021.

Li, Z., Xu, T., Qin, Z., Yu, Y., and Luo, Z.-Q. Imitation learning from imperfection: Theoretical justifications and algorithms. *Advances in Neural Information Processing Systems*, 36, 2024.

Ma, Y., Shen, A., Jayaraman, D., and Bastani, O. Versatile offline imitation from observations and examples via regularized state-occupancy matching. In *International Conference on Machine Learning*, pp. 14639–14663. PMLR, 2022.

Mandlekar, A., Zhu, Y., Garg, A., Booher, J., Spero, M., Tung, A., Gao, J., Emmons, J., Gupta, A., Orbay, E., et al. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In *Conference on Robot Learning*, pp. 879–893. PMLR, 2018.

Nie, W., Guo, B., Huang, Y., Xiao, C., Vahdat, A., and Anandkumar, A. Diffusion models for adversarial purification. In *International Conference on Machine Learning*, pp. 16805–16827. PMLR, 2022.

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

Pearce, T., Rashid, T., Kanervisto, A., Bignell, D., Sun, M., Georgescu, R., Macua, S. V., Tan, S. Z., Momennejad, I., Hofmann, K., et al. Imitating human behaviour with diffusion models. In *Deep Reinforcement Learning Workshop NeurIPS 2022*, 2022.

Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Series in Probability and Statistics. Wiley, 1994. ISBN 978-0-47161977-2. doi: 10.1002/9780470316887.

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

Särkkä, S. and Solin, A. *Applied stochastic differential equations*, volume 10. Cambridge University Press, 2019.

Sasaki, F. and Yamashina, R. Behavioral cloning from noisy demonstrations. In *International Conference on Learning Representations*, 2021.

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

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. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Song, Y., Durkan, C., Murray, I., and Ermon, S. Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems*, 34: 1415–1428, 2021a.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Tangkaratt, V., Han, B., Khan, M. E., and Sugiyama, M. Variational imitation learning with diverse-quality demonstrations. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 9407–9417, 2020.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

Torabi, F., Warnell, G., and Stone, P. Behavioral cloning from observation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 4950–4957, 2018.

Van Hasselt, H., Guez, A., and Silver, D. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

Wang, B., Wu, G., Pang, T., Zhang, Y., and Yin, Y. Diffail: Diffusion adversarial imitation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 15447–15455, 2024.

Wang, H.-C., Chen, S.-F., Hsu, M.-H., Lai, C.-M., and Sun, S.-H. Diffusion model-augmented behavioral cloning. In *ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems*, 2023a.

Wang, Y., Xu, C., and Du, B. Robust adversarial imitation learning via adaptively-selected demonstrations. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, pp. 3155–3161, 2021a.

Wang, Y., Xu, C., Du, B., and Lee, H. Learning to weight imperfect demonstrations. In *International Conference on Machine Learning*, pp. 10961–10970. PMLR, 2021b.

Wang, Y., Du, B., and Xu, C. Unlabeled imperfect demonstrations in adversarial imitation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 10262–10270, 2023b.

Wu, Y.-H., Charoenphakdee, N., Bao, H., Tangkaratt, V., and Sugiyama, M. Imitation learning from imperfect demonstration. In *International Conference on Machine Learning*, pp. 6818–6827. PMLR, 2019.

Xu, H., Zhan, X., Yin, H., and Qin, H. Discriminator-weighted offline imitation learning from suboptimal demonstrations. In *International Conference on Machine Learning*, pp. 24725–24742. PMLR, 2022.

Xu, T., Li, Z., and Yu, Y. Error bounds of imitating policies and environments. *Advances in Neural Information Processing Systems*, 33:15737–15749, 2020.

Xu, T., Li, Z., and Yu, Y. Error bounds of imitating policies and environments for reinforcement learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6968–6980, 2021.

Yu, L., Yu, T., Song, J., Neiswanger, W., and Ermon, S. Offline imitation learning with suboptimal demonstrations via relaxed distribution matching. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11016–11024, 2023.

Zha, D., Xie, J., Ma, W., Zhang, S., Lian, X., Hu, X., and Liu, J. Douzero: Mastering doudizhu with self-play deep reinforcement learning. In *International Conference on Machine Learning*, pp. 12333–12344. PMLR, 2021.

Zhang, S., Cao, Z., Sadigh, D., and Sui, Y. Confidence-aware imitation learning from demonstrations with varying optimality. *Advances in Neural Information Processing Systems*, 34:12340–12350, 2021.

Zhu, Y., Wong, J., Mandlekar, A., and Martín-Martín, R. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020.

A. Proof

A.1. Proof of Theorem 1

Theorem 1. *Let $\{x_t\}_{t\in\{0,1\}}$ be the forward diffusion process in Eq. 6. If we denote $\rho_{\pi_o,t}(x)$ and $\rho_{\pi_s,t}(x)$ as the respective distributions of x_t when $x_0 \sim \rho_{\pi_o,t=0}(x)$ and $x_0 \sim \rho_{\pi_s,t=0}(x)$, we then have,*

$$\varsigma \leq -\frac{1}{2}\int \rho_{\pi_o,t}(x)\beta_t||\nabla_x \log \rho_{\pi_o,t}(x) - \nabla_x \log \rho_{\pi_s,t}(x)||_2^2 \mathrm{d}x \leq 0. \tag{12}$$

where $\varsigma = \frac{\partial D_{KL}(\rho_{\pi_o,t}(x)||\rho_{\pi_s,t}(x))}{\partial t}$ denotes the derivative of t to the KL divergence between $\rho_{\pi_o,t}(x)$ and $\rho_{\pi_s,t}(x)$.

Proof. Following the proof in (Song et al., 2021a), we first make two assumptions on $\rho_{\pi_o,t}(x)$ and $\rho_{\pi_s,t}(x)$. Supposing both $\rho_{\pi_o,t}(x)$ and $\rho_{\pi_s,t}(x)$ are smooth and fast decaying functions, we have

$$\lim_{x\to\infty} \rho_{\pi_o,t}(x)\frac{\partial}{\partial x}\log \rho_{\pi_o,t}(x) = 0 \tag{13}$$

$$\lim_{x\to\infty} \rho_{\pi_s,t}(x)\frac{\partial}{\partial x}\log \rho_{\pi_s,t}(x) = 0 \tag{14}$$

The forward diffusion process in Eq. 6 is a discrete Markov chain, which can be written as

$$x_i = \sqrt{1-\beta_i}\cdot x_{i-1} + \sqrt{\beta_i}\cdot \epsilon_i, \quad i = 1, ..., N, \tag{15}$$

where $\epsilon_i \sim \mathcal{N}(0, I)$. To obtain a process of continuous transformation in time, we can rewrite Eq. 15 as

$$x_{(i+1)/N} = \sqrt{1-\frac{\bar{\beta}_{\frac{i+1}{N}}}{N}}x_{i/N} + \sqrt{\frac{\bar{\beta}_{\frac{i+1}{N}}}{N}}\epsilon_{(i+1)/N}, \quad i = 0, ..., N-1, \tag{16}$$

where $\{\bar{\beta}_{i/N} = N\beta_i\}_{i=0}^{N-1}$. In the limit of $N \longrightarrow \infty$, $\{\bar{\beta}_{i/N} = N\beta_i\}_{i=0}^{N-1}$ and $\{X_{i/N}\}_{i=0}^{N-1}$ become sequences of function $\{\beta_t\}_{t=0}^{1}$ and $\{x_t\}_{t=0}^{1}$. Let $\Delta t = \frac{1}{N}, t = \frac{i}{N}$, we can rewrite Eq. 16 as

$$x_{t+\triangle t} = \sqrt{1-\beta_{t+\triangle t}\cdot\triangle t}\cdot x_t + \sqrt{\beta_{t+\triangle t}\cdot\triangle t}\cdot\epsilon_t \tag{17}$$

$$= (1-\frac{1}{2}\beta_{t+\triangle t}\cdot\triangle t)\cdot x_t + \sqrt{\beta_{t+\triangle t}\triangle t}\cdot\epsilon_t + o(\beta_t\cdot\triangle t) \tag{18}$$

In the limit of $\triangle t \to 0$, we have

$$x_{t+\Delta t} - x_t = -\frac{1}{2}\beta_t\Delta t x_t + \sqrt{\beta_t\Delta t}\cdot\epsilon_t \tag{19}$$

In other words, we can transform Eq. 6 to the following VP-SDE,

$$\mathrm{d}x = -\frac{1}{2}\beta_t x\mathrm{d}t + \sqrt{\beta_t}\mathrm{d}w, \tag{20}$$

where w is a standard Wiener process. The Fokker–Planck equation for the VP-SDE above describes the time-evolution of the stochastic process's associated probability density function, and is given by

$$\frac{\partial \rho_{\pi_o,t}(x)}{\partial t} = -\nabla_x\cdot\Big(\frac{1}{2}\beta_t\rho_{\pi_o,t}(x)\nabla_x\log\rho_{\pi_o,t}(x) + \frac{1}{2}\beta_t x\rho_{\pi_o,t}(x)\Big) \tag{21}$$

$$= -\nabla_x\cdot\Big(h_o(x,t)\rho_{\pi_o,t}(x)\Big), \tag{22}$$

where $h_o(x,t)$ is defined as $h_o(x,t) = \frac{1}{2}\beta_t\nabla_x\log\rho_{\pi_o,t}(x) + \frac{1}{2}\beta_t x$. Then, we denote $\varsigma = \frac{\partial D_{KL}(\rho_{\pi_o,t}(x)||\rho_{\pi_s,t}(x))}{\partial t}$ and ς

can be re-written as follows,

$$\varsigma = \frac{\partial}{\partial t}\int \rho_{\pi_o,t}(x)\log\frac{\rho_{\pi_o,t}(x)}{\rho_{\pi_s,t}(x)}\mathrm{d}x \tag{23}$$

$$= \int \frac{\partial \rho_{\pi_o,t}(x)}{\partial t}\log\frac{\rho_{\pi_o,t}(x)}{\rho_{\pi_s,t}(x)}\mathrm{d}x + \int \frac{\partial \rho_{\pi_o,t}(x)}{\partial t}\mathrm{d}x - \int \frac{\rho_{\pi_o,t}(x)}{\rho_{\pi_s,t}(x)}\frac{\partial \rho_{\pi_s,t}(x)}{\partial t}\mathrm{d}x \tag{24}$$

$$\overset{(i)}{=} \int \nabla_x \cdot \Big(- h_o(x,t)\rho_{\pi_o,t}(x)\Big)\log\frac{\rho_{\pi_o,t}(x)}{\rho_{\pi_s,t}(x)}\mathrm{d}x - \int \frac{\rho_{\pi_o,t}(x)}{\rho_{\pi_s,t}(x)}\nabla_x \cdot \Big(- h_s(x,t)\rho_{\pi_s,t}(x)\Big) \tag{25}$$

$$\overset{(ii)}{=} - \int \rho_{\pi_o,t}(x)[h_o^T(x,t) - h_s^T(x,t)][\nabla_x \log \rho_{\pi_o,t}(x) - \nabla_x \log \rho_{\pi_s,t}(x)] \tag{26}$$

$$= -\frac{1}{2}\int \rho_{\pi_o,t}(x)\beta_t|\nabla_x \log \rho_{\pi_o,t}(x) - \nabla_x \log \rho_{\pi_s,t}(x)|_2^2\mathrm{d}x \tag{27}$$

$$= -\frac{\beta_t}{2}E_{x\sim\rho_{\pi_o,t}(x)}(|\nabla_x \log \rho_{\pi_o,t}(x) - \nabla_x \log \rho_{\pi_s,t}(x)|_2^2) \tag{28}$$

$$\leq \quad 0 \tag{29}$$

where (i) can be obtained by Eq. 13 and Eq. 14, (ii) can be obtained by integration by parts. □

A.2. Proof of Theorem 2

Theorem 2. *Supposing the sub-optimal demonstration x_s perturbed by noise δ compared to the optimal demonstration x_o, and $\hat{x}_o$ is a purified demonstration of x_s, the L2 distance between x_o and $\hat{x}_o$ (i.e., $\|x_o - \hat{x}_o\|$) is bounded within the interval $[\max\{0,\lambda\},\Lambda]$ with a probability of at least $1-\sigma$, where*

$$\lambda(\zeta, t_r, \delta, \sigma) = \zeta(t_r)\tilde{C}_{gd} - \|\delta\| - \sqrt{e^{2\zeta(t_r)} - 1}C_{\sigma,d}, \tag{30}$$

$$\Lambda(\zeta, t_r, \delta, \sigma) = \zeta(t_r)C_{gd} + \|\delta\| + \sqrt{e^{2\zeta(t_r)} - 1}C_{\sigma,d}, \tag{31}$$

and $\tilde{C}_{gd} \leq 2\|\nabla_x \log \rho_t(x)\| \leq C_{gd}$, $\zeta(t_r) = \int_0^{t_r}\frac{1}{2}\beta_s\mathrm{d}s$, $C_{\sigma,d} = \sqrt{2d(1 + \sqrt{\frac{8\ln\frac{2}{\sigma}}{d}})}$, $\sigma \geq 2e^{-d/8}$.

Proof. Following the proof in (Nie et al., 2022), we extend the distance bound to both upper and lower bounds. The sub-optimal demonstration x_s can be conceptualized as a perturbation of the optimal demonstration x_o, with a small disturbance δ applied to each state-action pair. Furthermore, let x_t denote the demonstration obtained after subjecting x_s to a forward diffusion process satisfies

$$x_t = \sqrt{\gamma(t)}x_s + \sqrt{1-\gamma(t)}\epsilon_1, \tag{32}$$

where $\gamma(t) = \exp(-\int_0^t \beta_s\mathrm{d}s)$ and $\epsilon_1 \in \mathcal{N}(0, I_d)$, the L2 distance between the purified demonstrations $\hat{x}_o$ and its related optimal demonstration x_o can be upper bounded as

$$\|\hat{x}_o - x_o\| = \|x_t + (\hat{x}_o - x_t) - x_o\| \tag{33}$$

$$= \|x_t + \int_t^0 -\frac{1}{2}\beta_t[x + 2\nabla_x \log \rho_t(x)]\mathrm{d}t + \int_t^0 \sqrt{\beta_t}\mathrm{d}\overline{w} - x_o\| \tag{34}$$

$$\leq \|x_t + \int_t^0 -\frac{1}{2}\beta_t x\mathrm{d}t + \int_t^0 \sqrt{\beta_t}\mathrm{d}\overline{w} - x\| + \|\int_t^0 -\beta_t\nabla_x \log \rho_t(x)\mathrm{d}t\|, \tag{35}$$

where Eq. 34 follows with the definition of reverse-time diffusion and Eq. 35 can be obtained via triangle inequality. Similarly, it can be lower bounded as

$$\|\hat{x}_o - x_o\| \geq \|\int_t^0 \beta_t\nabla_x \log \rho_t(x)\mathrm{d}t\| - \|x_t + \int_t^0 -\frac{1}{2}\beta_t x\mathrm{d}t + \int_t^0 \sqrt{\beta_t}\mathrm{d}\overline{w} - x\| \tag{36}$$

The sum of the first 3 terms in Eq. 34 is a time-varying Ornstein-Uhlenbeck process with a negative time increment that starts from $t = t$ to $t = 0$ with the initial value set to x_t. Denote by x_0' its solution, from (Särkkä & Solin, 2019) we know

x_0' follows a Gaussian distribution, where its mean $\mu(0)$ and covariance matrix $\Sigma(0)$ are the solutions of the following two differential equations, respectively,

$$\frac{\mathrm{d}\mu}{\mathrm{d}t} = -\frac{1}{2}\beta_t\mu, \quad \frac{\mathrm{d}\Sigma}{\mathrm{d}t} = -\beta_t\Sigma + \beta_t I_d, \tag{37}$$

with the initial conditions $\mu(t) = x_t$ and $\Sigma(t) = 0$. By solving these two differential equations, we have that conditioned on x_t, $x_0' \sim \mathcal{N}(e^{\zeta(t_r)}x_t, (e^{2\zeta(t_r)} - 1)I_d)$, where $\zeta(t_r) = \int_0^t \frac{1}{2}\beta_s \mathrm{d}s$.

Note that,

$$x_t = \sqrt{\gamma(t)}x_s + \sqrt{1-\gamma(t)}\epsilon_1 \tag{38}$$

$$= e^{-\zeta(t_r)}(x_o + \delta) + \sqrt{1 - e^{-2\zeta(t_r)}}\epsilon_1 \tag{39}$$

Using the reparameterization trick, we have,

$$x_0' - x_o = e^{\zeta(t_r)}x_t + \sqrt{e^{2\zeta(t_r)} - 1}\epsilon_2 - x_o \tag{40}$$

$$= e^{\zeta(t_r)}\Big(e^{-\zeta(t_r)}(x_o + \delta) + \sqrt{1 - e^{-2\zeta(t_r)}}\epsilon_1\Big) + \sqrt{e^{2\zeta(t_r)} - 1}\epsilon_2 - x_o \tag{41}$$

$$= \sqrt{e^{2\zeta(t_r)} - 1}(\epsilon_1 + \epsilon_2) + \delta \tag{42}$$

$$= \sqrt{2(e^{2\zeta(t_r)} - 1)}\epsilon + \delta \tag{43}$$

Since ϵ_1 and ϵ_2 are independent and taken from the distribution $\mathcal{N}(0, I_d)$, $\epsilon = \frac{\epsilon_1 + \epsilon_2}{\sqrt{2}} \sim \mathcal{N}(0, I_d)$. Assuming that $\tilde{C}_{gd} \leq 2\|\nabla_x \log \rho_t(x)\| \leq C_{gd}$, we have,

$$\|\hat{x}_o - x_o\| \leq \|x_0' - x_o\| + \|\int_{t_r}^0 -\beta_t \nabla \log \rho_t(x))\mathrm{d}t\| \tag{44}$$

$$= \Big\|\sqrt{2(e^{2\zeta(t_r)} - 1)}\epsilon + \delta\Big\| + \zeta(t_r)C_{gd} \tag{45}$$

$$\leq \Big\|\sqrt{2(e^{2\zeta(t_r)} - 1)}\epsilon\Big\| + \|\delta\| + \zeta(t_r)C_{gd} \tag{46}$$

Similarly, we also have,

$$\|\hat{x}_o - x\| \geq \|\int_{t_r}^0 -\beta_t \nabla \log \rho_t(x)\mathrm{d}t\| - \|x_0' - x\| \tag{47}$$

$$\geq \zeta(t_r)\tilde{C}_{gd} - \Big\|\sqrt{2(e^{2\zeta(t_r)} - 1)}\epsilon + \delta\Big\| \tag{48}$$

$$\geq \zeta(t_r)\tilde{C}_{gd} - \Big\|\sqrt{2(e^{2\zeta(t_r)} - 1)}\epsilon\Big\| - \|\delta\| \tag{49}$$

Since $\|\epsilon\|^2 = \mathbf{Z}_1^2 + \mathbf{Z}_2^2 + \cdots + \mathbf{Z}_d^2 \sim \mathcal{X}^2(d)$, where $\mathbf{Z}_i^2 \sim \mathcal{N}(0, 1)$. It can be regarded as a sub-exponential random variable with parameters $(\nu, \alpha) = (2\sqrt{d}, 4)$, we have

$$Pr\Big(|\frac{1}{d}\sum_{n=1}^{d}\mathbf{Z}_n^2 - 1| \geq t\Big) \leq 2e^{-dt^2/8}, \quad \text{for all} \quad t \in (0, 1) \tag{50}$$

Let $\sigma = 2e^{-dt^2/8}$, we have

$$Pr\left(\sqrt{d(1 - \sqrt{\frac{8ln(2/\sigma)}{d}})} \leq \|\epsilon\| \leq \sqrt{d(1 + \sqrt{\frac{8ln(2/\sigma)}{d}})}\right) \geq 1 - \sigma \tag{51}$$

By integrating the results above, we can derive Eqs. 30, 31. These two equations present the lower and upper bounds of the L2 distance between $\hat{x}_o$ and x_o with the probability of at least $1 - \sigma$, where $\sigma \geq 2e^{-d/8}$. □

A.3. Proof of Proposition 3

Proposition 3. *$\lambda(\zeta, t_r, \delta, \sigma)$ and $\Lambda(\zeta, t_r, \delta, \sigma)$ are monotonically increasing with respect to t_r.*

Proof. The derivative of $\lambda(t_r)$ with respect to t_r can be written as

$$\lambda'(t_r) = -\zeta'(t_r)(e^{2\zeta(t_r)} - 1)^{-1/2} C_{\sigma,d} + \zeta'(t_r)\tilde{C}_{gd} \tag{52}$$

With the assumption that the lower bound is non-negative, we have

$$\tilde{C}_{gd} \geq \frac{1}{\zeta(t_r)}(\|\delta\| + \sqrt{e^{2\zeta(t_r)} - 1} C_{\sigma,d}) \tag{53}$$

By substituting Eq. 53 into Eq. 52, we have,

$$\lambda'(t_r) \geq \frac{\zeta'(t_r)}{\zeta(t_r)}(e^{2\zeta(t_r)} - 1)^{-1/2}[C_{\sigma,d}(e^{2\zeta(t_r)} - \zeta(t_r) - 1) + \sqrt{e^{2\zeta(t_r)} - 1}\|\delta\|] \tag{54}$$

Given that $\zeta(t_r) \geq 0$, $\zeta'(t_r) = \beta_{t_r} \geq 0$ and $e^{2\zeta(t_r)} - \zeta(t_r) - 1 \geq 0$, it follows that $\lambda'(t_r) \geq 0$. Considering the upper bound $\Lambda(t_r) = \zeta(t_r)C_{gd} + \|\delta\| + \sqrt{e^{2\zeta(t_r)} - 1} C_{\sigma,d}$, and noting that $\Lambda(t_r)$ increases monotonically as a function of $\zeta(t_r)$, and since $\zeta(t_r)$ itself increases monotonically with t_r, it can be deduced that $\Lambda(t_r)$ is also a monotonically increasing function of t_r. □

A.4. Proof of Theorem 4

Theorem 4. *There exists a inflection point t_p that satisfies $\lambda''(t_p) = 0$, which makes $\lambda(t_r)$ increases smoothly before t_p while increasing rapidly after t_p. The optimal t_r^* is around the inflection point t_p, or is around the solution of the equation*

$$-\frac{(\zeta'(t_r))^2}{\zeta''(t_r)}(e^{2\zeta(t_r)} - 1)^{-3/2} + (e^{2\zeta(t_r)} - 1)^{-1/2} = \frac{\tilde{C}_{gd}}{C_{\sigma,d}}. \tag{55}$$

With a probability of $1 - \sigma$, there is a positive correlation between $\|\delta\|$ and t_r^*.

Proof. We aim to identify a critical point t_p where the rate of growth of the function experiences a significant increase. In mathematical words, $\lambda(t_r)$ should satisfies $\lambda'(t_r) \geq 0$, $\lambda''(t_p) = 0$ and $\lambda''(t_r) < 0$ for $t_r < t_p$, $\lambda''(t_r) > 0$ for $t_r > t_p$. The second derivative of $\lambda(t_r)$ can be expressed as:

$$\lambda''(t_r) = \left[-\zeta''(t_r) + (\zeta'(t_r))^2(e^{2\zeta(t_r)} - 1)\right](e^{2\zeta(t_r)} - 1)^{-1/2} C_{\sigma,d} + \zeta''(t_r)\tilde{C}_{gd} \tag{56}$$

Therefore, $\lambda''(t_r) = 0$ if and only if Eq. 55 holds.

Now, we can analyze the correlation between $\|\delta\|$ and t_r^*. Note that $\tilde{C}_{gd}$ is the lower bound of $2\|\nabla_x \log \rho_t(x)\|$. With the increase of disturbance $\|\delta\|$, $\tilde{C}_{gd}$ will subsequently increase, causing the right-hand side of Eq. 55 to increase.

In our setting, $\zeta(t_r) = \frac{1}{4}kt_r^2$. Therefore, Eq. 55 can be rewritten as:

$$-\frac{1}{2}kt_r^2(e^{\frac{1}{2}kt_r^2} - 1)^{-3/2} + (e^{\frac{1}{2}kt_r^2} - 1)^{-1/2} = -2\zeta(t_r)(e^{2\zeta(t_r)} - 1)^{-3/2} + (e^{2\zeta(t_r)} - 1)^{-1/2} = \frac{\tilde{C}_{gd}}{C_{\sigma,d}} \tag{57}$$

The left-hand side of Eq. 55 is monotonically increasing with respect to $\zeta(t_r)$ in the interval $[0, 1]$. Since $\zeta(t_r)$ is also a monotonically increasing function with respect to t_r, within a certain range, the left-hand side of the equation is a monotonically increasing function with respect to t_r. In other words, an increase in $\|\delta\|$ will lead to an increase in t_r^*, indicating a positive correlation between $\|\delta\|$ and t_r^*. □

A.5. Sensitive Analysis of σ

The lower bound of $\|x_o - \hat{x_o}\|$ can be regarded as a function with four parameters: t_r, σ, d, and δ. However, once the optimal demonstration x_o and the corresponding sub-optimal demonstration x_s are provided, δ and d are fixed. Furthermore, considering that different demonstrations exhibit varying tolerances for sub-optimal behavior, the magnitude of σ can be set according to actual scenarios. In most cases, $\sigma \in [0.05, 0.0001]$. Therefore, λ is indeed an increasing function with respect to t_r when a pair of demonstrations and a confidence level $1 - \sigma$ are specified. However, it can be observed that among these four parameters, only σ is subjectively determined by humans. We hope that for the same pair of optimal and sub-optimal demonstrations, the different subjective choices of each individual have little impact on the final calculated value of t_r. We provide a new theorem below to show that different confidence level σ has little impact on the choice of t_r.

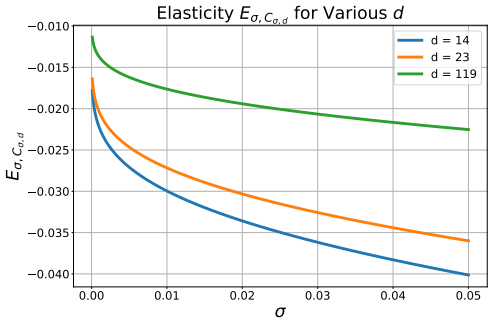

Figure 4. The corresponding curve of the elasticity $E_{\sigma,C_{\sigma,d}}$ with respect to σ.

Theorem 5. *The selection of t_r^* exhibits low sensitivity to the magnitude of σ, indicating that, for a purified demonstration, configuring different confidence levels has minimal impact on t_r^*. Additionally, as the dimension of the demonstration d increases, the sensitivity of t_r^* to σ will also decrease.*

Proof. First of all, we will introduce elasticity, which is an indicator that can be used to measure the extent to which a change in one variable will affect other variables. Define elasticity $E_{x,y}$ of variables x and y as follows,

$$E_{x,y} = \frac{d(lny)}{d(lnx)}. \tag{58}$$

We have already known that $C_{\sigma,d} = \sqrt{2d(1 + \sqrt{\frac{8ln(2/\sigma)}{d}})}$, then

$$E_{\sigma,C_{\sigma,d}} = -\frac{1}{\sqrt{2dln(2/\sigma)} + 4ln(2/\sigma)}. \tag{59}$$

Since $1 - \sigma$ represents the confidence level, in most cases, the value range of σ is 0.05 to 0.0001. Note that the higher the accuracy requirement for the demonstration, the smaller the corresponding selected σ. In our experiment, the dimension of data is 14, 23 and 119, and the corresponding curve of the elasticity $E_{\sigma,C_{\sigma,d}}$ is in Figure 4.

From the figure, we can find that $\|E_{\sigma,C_{\sigma,d}}\| < 0.0113$ for every d, which reflects that $C_{\sigma,d}$ is not sensitive to the change of σ , when the value range of σ is 0.05 to 0.0001. In light of the analysis derived from Eq. 59, it is evident that with increasing complexity in the demonstration, as indicated by the augmentation of the data dimension d, the sensitivity of parameter $E_{\sigma,C_{\sigma,d}}$ to variations in parameter σ exhibits a notable decrement. This observation implies that in scenarios involving more intricate demonstrations, particularly when the value of σ lies within the range of 0.05 to 0.0001, the selection of σ becomes substantially less consequential. This diminishing relevance of σ selection is a critical consideration in the context of increasingly complex data dimensions, as it underscores when dealing with a high-dimensional real-world data, the choice of different σ has a negligible impact on the optimal t_r^*.

□

B. Experimental Details

Our source code and training data will be available at https://github.com/yunke-wang/dp-il.

B.1. Implementation Details of Diffusion Model

The architecture of diffusion model ϵ_ϕ is five linear layers with a 0.2 dropout ratio, batch normalization, and ReLU nonlinear activation, and the size of the hidden dimension is 1024. We implement the above two algorithms based on this repo (https://github.com/abarankab/DDPM). The training epoch is set to be 10000 and the learning rate of ϵ_ϕ is set to 1e-4. We set $N = 1000$ for all experiments and set the forward process variances to constants increasing linearly from $\beta_1 = 1e - 4$ to

$\beta_N = 0.02$. The pseudo code of diffusion model's training and purification is available in Algorithm 1 and Algorithm 2.

Algorithm 1 Diffusion Model Training

repeat
  Sample optimal demonstrations $x_o \sim \mathcal{D}_o$
  $i \sim \text{Uniform}(\{0, ..., N-1\})$
  $\epsilon \sim \mathcal{N}(0, I)$
  Optimize diffusion model ϵ_ϕ by taking gradient descent step on
  $\nabla_\phi[\epsilon - \epsilon_\phi(\sqrt{\bar{\alpha}_i} \cdot x_o + \sqrt{1-\bar{\alpha}_i} \cdot \epsilon, i)]^2$
until converged

Algorithm 2 Diffusion Purification

Sample sub-optimal demonstrations $x_s \sim \mathcal{D}_s$;
Calculate x_{s,i^*} via forward diffusion:
$x_{s,i^*} = \sqrt{\bar{\alpha}_i} \cdot x_s + \sqrt{1-\bar{\alpha}_i} \cdot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$.
for $i = i^*, ..., 1$ **do**
  Calculate x_{i-1} via reverse diffusion:
  $x_{s,i-1} = \frac{1}{\sqrt{\alpha_i}}\left(x_{s,i} - \frac{1-\alpha_i}{\sqrt{1-\bar{\alpha}_i}\epsilon_\phi(x_{s,i}, t)}\right) + \sqrt{\beta_i} z,$
end for
Return $\hat{x}_{s,0}$

B.2. Implementation Details of Imitation Learning

We use a deep neural network that has two 100×100 fully connected layers and uses Tanh as the activation layer to parameterize policy. To output continuous action, agent policy adopts a Gaussian strategy, hence the policy network outputs the mean and standard deviation of action. The continuous action is sampled from the normal distribution formulated with the action's mean and standard deviation. For online imitation learning methods, the discriminator and value function are using the same architecture as the policy network.

In offline imitation learning, the policy is trained with batch size 256, and the total epoch is set to be 1000. For online imitation learning, the learning rate of the discriminator D_ψ and the critic r_ψ is set to 3×10^{-4}. Five updates on the discriminator follow with one update on the policy network in one iteration. For the value function, the learning rate is set to 3×10^{-4} and three training updates are used in one iteration. We conduct the on-policy method TRPO (Schulman et al., 2015) as RL step in online imitation learning, the learning rate is set to 3×10^{-4} with batch size 5000. The discount rate γ of the sampled trajectory is set to 0.995. The τ (GAE parameter) is set to 0.97.

B.3. Data Collection

We provide six supplementary sub-optimal demonstration datasets to evaluate the performance of DP-IL on 4 MuJoCo tasks. After we train an optimal policy π_o by TRPO, we use **two** different kinds of ways to collect sub-optimal demonstration data in our experiments. The quality of sub-optimal demonstrations is provided in Table 6.

The first way to collect sub-optimal demonstrations is to add different Gaussian noise ξ to the action distribution a^* of π_s to form sub-optimal policy π_s. The action of π_s is modeled as $a \sim \mathcal{N}(a^*, \xi^2)$ and we choose $\xi = [0.6, 0.4, 0.25]$ to generate sub-optimal demonstrations datasets (*e.g.*, D1-L1, D1-L2 and D1-L3). This way of collecting sub-optimal demonstration has been used in several works (Sasaki & Yamashina, 2021; Tangkaratt et al., 2020).

The second way to collect sub-optimal demonstrations is to save checkpoints during the RL training of $\pi_s(a|s)$. In our experiment, the RL training is conducted with 5M interactions with the environment. We save 3 checkpoints at 1M, 1.5M and 3M interactions to sample D2-L1, D2-L2 and D2-L3 datasets. While the agent converges fast This way of collecting sub-optimal demonstration has been investigated in (Wang et al., 2021b; Wu et al., 2019).

Table 6. The quality of demonstrations in 4 MuJoCo tasks, which is measured by the average cumulative reward of trajectories.

Task	$\mathcal{S}$	$\mathcal{A}$	D1-L1	D1-L2	D1-L3	D2-L1	D2-L2	D2-L3	**Expert**
Ant-v2	$\mathbb{R}^{111}$	$\mathbb{R}^8$	-73	227	3514	1062	1560	2649	4349
HalfCheetah-v2	$\mathbb{R}^{17}$	$\mathbb{R}^6$	567	1090	1853	1491	2217	3263	4624
Walker2d-v2	$\mathbb{R}^{17}$	$\mathbb{R}^6$	523	467	4362	1699	2717	4152	4963
Hopper-v2	$\mathbb{R}^{11}$	$\mathbb{R}^3$	699	1037	3229	734	3362	2597	3594

B.4. Compared Methods

We compare DP-IL with several *state-of-the-art* imitation learning with imperfect demonstration methods. Specifically, several offline imitation learning methods (*i.e.*, BCND (Sasaki & Yamashina, 2021), DWBC (Xu et al., 2022) and DemoDICE (Kim et al., 2021)) and online imitation learning methods (*i.e.*, 2IWIL (Wu et al., 2019), IC-GAIL (Wu et al., 2019) and WGAIL (Wang et al., 2021b)) are compared. We briefly review the details of methods compared against DP-IL in our experiments below.

BCND, DWBC and DemoDICE We follow the instruction in BCND's paper to implement BCND. Notice that for a fair comparison with other offline imitation learning methods, we do not use ensemble policies for BCND. As for DWBC, we adapt their released code[1] to conduct experiments. As for DemoDICE, we adapt the implementation in SmoDICE repo[2] to conduct experiments.

2IWIL and IC-GAIL We re-implement 2IWIL/IC-GAIL based on the official implementation[3]. In 2IWIL and IC-GAIL, a fraction of imperfect expert demonstrations are labeled with confidence (*i.e.*, $\mathcal{D}_l = \{(s_i, a_i), r_i\}_i^{n_l}$), while the remaining demonstrations are unlabeled (*i.e.*, $\mathcal{D}_u = \{(s_i, a_i)\}_i^{n_u}$). Since we have no access to the confidence score of the state-action pair in our setting, we use the normalized reward of the demonstrator as the confidence score for their related demonstrations. In the experiment, we choose 20% labeled demonstrations to train a semi-supervised classifier and then predict confidence for other 80% unlabeled demonstrations.

WGAIL WGAIL is proposed to estimate confidence in GAIL framework without auxiliary information. The confidence $w(s, a)$ of each demonstration is calculated by $[(1/D_w^*(s, a) - 1)\pi_\theta(a|s)]^{\frac{1}{\beta+1}}$. The confidence estimation and GAIL training interact during the training. Followed with the official implementation[4], β is set to be 1.

B.5. Additional Experimental Results

B.5.1. Impact of reverse point i_r in RoboSuite

We also include Robosuite results to offer a more complete understanding of how i_r^* impacts demonstrations of varying quality. In Robosuite, we utilized 10 trajectories as expert demonstrations, as shown in Table 7. Among these, two high-reward trajectories (ID 6 and 10) were considered optimal demonstrations, while the rest were designated as sub-optimal demonstrations. To assess the impact of different demonstration qualities, we divided the sub-optimal demonstrations into two groups, Robo-L1 and Robo-L2.

Table 7. The length and quality of 10 trajectories in RoboSuite.

Traj_ID	1	2	3	4	5	6	7	8	9	10
Length	500	502	502	505	506	507	507	509	511	511
Reward	276	120	203	102	253	312	103	81	180	318

For Robo_L1, we selected four trajectories with rewards below 150 (ID 2, 4, 7, and 8) as sub-optimal demonstrations. For Robo-L2, we chose four trajectories with rewards ranging from 150 to 300 (ID 1, 3, 5, and 9) as sub-optimal demonstrations. Clearly, Robo-L2 demonstrates higher quality compared to Robo-D1. We assessed the influence of timestep i_r^* on both Robo-L1 and Robo-L2. The results, as shown in Table 8, exhibit similar trends to those observed in the MuJoCo tasks.

Table 8. Impact of i_r in RoboSuite platform with human demonstrations.

i_r	1	3	5	10	30
DP-BC (Robo-L1)	0.60	0.74	**0.80**	0.64	0.48
DP-BC (Robo-L2)	**0.82**	0.78	0.76	0.76	0.52

[1] https://github.com/ryanxhr/DWBC
[2] https://github.com/JasonMa2016/SMODICE/tree/main
[3] https://github.com/kristery/Imitation-Learning-from-Imperfect-Demonstration
[4] https://github.com/yunke-wang/WGAIL

B.6. Impact of reverse point i_r in MuJoCo

We have provided the ablation study on i_r with D1 demonstrations in Figure 2, and we provide full results of i_r with both D1 and D2 demonstrations in Table 9. From the table, we can observe that in most cases (19 out of 24 cases), setting i_r within the 5-10 range consistently leads to optimal or near-optimal performance. These cases are across various tasks and demonstration qualities, which indicates that an i_r value in this range can serve as a robust default choice that yields satisfactory results in most situations.

Table 9. Performance of DP-BC when setting different i_r, which is measured by the average cumulative reward of 10 trajectories. The results are related to Figure 2.

Task	i_r	D1-L1	D1-L2	D1-L3	D2-L1	D2-L2	D2-L3
Ant-v2	100	**261±54**	380±62	486±76	199±66	342±14	328±88
	50	202±51	**803±114**	1073±172	339±27	720±139	1776±123
	30	47±16	719±51	1890±156	**1572±160**	1833±30	3064±34
	10	-100±51	499±77	**2547±118**	1402±151	1982±111	3260±34
	5	-133±55	490±125	2331±248	1105±263	**1984±103**	**3414±40**
HalfCheetah-v2	100	556±102	640±108	1115±181	1123±41	1791±13	1393±175
	50	797±185	1896±222	2733±113	1468±82	2421±5	3126±306
	30	895±73	1880±326	2899±257	**1530±39**	2714±15	4434±240
	10	**1365±147**	**2440±274**	**4042±80**	1394±103	**2749±15**	**4748±40**
	5	708±255	2061±373	3758±351	1469±253	2469±253	4660±14
Walker2d-v2	100	260±2	241±12	311±9	259±1	258±1	273±19
	50	463±40	259±11	553±89	549±196	360±14	1085±251
	30	415±56	978±96	1398±333	**2208±160**	1742±253	2512±212
	10	**1697±219**	1267±145	1474±265	1764±143	1915±384	1984±365
	5	1467±253	**1722±297**	**3020±466**	1917±151	**3162±20**	**3076±205**
Hopper-v2	100	98±2	461±11	420±1	215±1	151±3	355±25
	50	516±0	815±16	379±0	1283±109	631±4	342±29
	30	688±33	649±43	517±1	2075±159	904±6	1353±128
	10	**1000±42**	2752±38	678±56	**2308±113**	**2323±2**	1740±175
	5	580±60	**3145±11**	**1508±272**	1835±216	2297±34	**2265±171**

While in most cases, an i_r setting within the range of 5-10 yields optimal performance, we can also observe the selection of i_r has a connection to the task. In the 4 MuJoCo tasks used in the experiment, Ant-v2 is the most challenging task, followed by HalfCheetah-v2 and Walker2d-v2, with Hopper-v2 being the easiest. We can observe that the average optimal i_r is relatively higher (for instance, around 100 for D1-L1) for Ant-v2 and smaller for Hopper-v2, where an optimal i_r is as low as 5 for all datasets. This suggests that setting a relatively higher i_r for more difficult tasks is reasonable. Following the results presented in Table 9, an empirical way is to set i_r from 5 to 10 for simple tasks like Hopper-v2, and set i_r from 10 to 50 for challenging tasks like Ant-v2.