Sample-efficient Adversarial Imitation Learning

Dahuin Jung   ANNAJUNG0625@SNU.AC.KR
Electrical and Computer Engineering
Seoul National University, Seoul 08826, Republic of Korea
Hyungyu Lee   RUCY74@SNU.AC.KR
Electrical and Computer Engineering
Seoul National University, Seoul 08826, Republic of Korea
Sungroh Yoon[*]   SRYOON@SNU.AC.KR
Electrical and Computer Engineering
Interdisciplinary Program in Artificial Intelligence
Seoul National University, Seoul 08826, Republic of Korea

Editor: Scott Niekum

Abstract

Imitation learning, in which learning is performed by demonstration, has been studied and advanced for sequential decision-making tasks in which a reward function is not predefined. However, imitation learning methods still require numerous expert demonstration samples to successfully imitate an expert's behavior. To improve sample efficiency, we utilize self-supervised representation learning, which can generate vast training signals from the given data. In this study, we propose a self-supervised representation-based adversarial imitation learning method to learn state and action representations that are robust to diverse distortions and temporally predictive, on non-image control tasks. In particular, in comparison with existing self-supervised learning methods for tabular data, we propose a different corruption method for state and action representations that is robust to diverse distortions. We theoretically and empirically observe that making an informative feature manifold with less sample complexity significantly improves the performance of imitation learning. The proposed method shows a 39% relative improvement over existing adversarial imitation learning methods on MuJoCo in a setting limited to 100 expert state-action pairs. Moreover, we conduct comprehensive ablations and additional experiments using demonstrations with varying optimality to provide insights into a range of factors.

Keywords: imitation learning, adversarial imitation learning, self-supervised learning, data efficiency

1. Introduction

Imitation learning (IL) is widely used in sequential decision-making tasks, where the design of a reward function is complicated or uncertain. When a reward is sparse (Reddy et al., 2019) or an optimal reward function is unknown, IL finds an optimal policy that relies only on expert demonstrations. Owing to recent development in deep neural networks, the range

---

*. Corresponding Author
    A preliminary version of this manuscript was presented at Deep RL Workshop, NeurIPS 2022.

of behaviors, which can be imitated, has expanded. There are two main learning approaches for IL. The first approach trains a policy by following actions from an expert in a supervised manner called behavioral cloning (BC) (Sutton and Barto, 1998; Pomerleau, 1989). However, error accumulation limits BC because it greedily imitates the demonstrated actions. For behavior cloning, it talks about the problem of accumulating errors, but there are alternative IL approaches like DAgger that deals with that. The second approach is inverse reinforcement learning (IRL) (Arora and Doshi, 2021), inferring a cost function based on given expert demonstrations. The IRL implements adversarial learning (Goodfellow et al., 2014) to infer the cost function. Therefore, an agent learns the policy to imitate expert demonstrations, whereas a discriminator learns to differentiate between the expert's behavior and that of the agent. The learned discriminator is used as the cost function in the reinforcement learning (RL) phase. However, there are also several IRL methods that do not leverage adversarial training, e.g., Max-Margin Planning (Ratliff et al., 2006) and Max-Ent IRL (Ziebart et al., 2008b).

Although IRL has led an advance in IL, it has key challenges. First, adversarial learning is known to be delicate in practice. The min-max computational formulation of adversarial imitation learning (AIL) often involves brittle approximation techniques. Second, the IL requires many demonstration trajectories to recover an expert policy. Although IRL requires fewer demonstrations than BC, it still requires considerable trajectories. Recently, many algorithms or techniques have been proposed to address the first challenge (Fu et al., 2018; Peng et al., 2019; Reddy et al., 2019; Qureshi et al., 2018; Chen et al., 2019; Wang et al., 2022); however, little work has been done to improve the sample efficiency of expert demonstrations required (Zhang et al., 2020; Barde et al., 2020).

Self-supervised representation learning (SSL) has advanced sample efficiency in the image and language domains (Chen et al., 2020b; Grill et al., 2020; Mikolov et al., 2013). It applies various transformations to the data and uses the transformed data itself as supervision. Thus, it increases the sample efficiency by obtaining training signals from auxiliary tasks or objectives that do not rely on labels. Specifically, InfoNCE (Chen et al., 2020b; He et al., 2020) and asymmetric twin-based (Grill et al., 2020; Chen and He, 2021) SSL approaches are known to be effective for learning robust feature representations for different distortions of identical inputs. Recently, SSL has been utilized in image-based RL algorithms (Srinivas et al., 2020; Schwarzer et al., 2020; Li et al., 2022) and has shown significant improvement in performance. However, transformation techniques applied to image-based RL are not directly adaptable to non-image control benchmarks. This is because these approaches rely upon the semantic/spatial properties of data that may generate either out-of-distribution examples or examples that supply only the same view when directly applied to a continuous control (tabular) domain.

In this study, we propose a sample-efficient AIL method for non-image control benchmarks. The proposed method leverages auxiliary training signals for learning state and action representations that are temporally predictive and robust to diverse distortions. Based on the characteristics of each domain and benchmark, an auxiliary task that can learn informative feature representations is different. For RL, to address sequential decision-making tasks, the feature representation of a state and action should contain temporally predictive information. To address this, we add an auxiliary task that predicts the next state representation from the given current state and action representations. This marks the initial attempt to enhance

the sample efficiency of expert demonstrations in non-image imitation learning by employing SSL.

Moreover, learning representations that discard information regarding nuisance variables improves generalization and decreases required sample complexity. Previous transformation techniques for tabular data (Yoon et al., 2020; Bahri et al., 2021) generate transformed samples far from real samples. Therefore, we propose a simple, effective corruption method that generates transformed samples showing diverse distortions that are possible in-distribution. Empirically, we demonstrate that promoting temporally predictive feature representations with robustness against diverse distortions significantly improves sample efficiency.

First, we theoretically observe that IL performs substantially better when an informative feature manifold is created with less sample complexity. We evaluate the proposed method on MuJoCo (Todorov et al., 2012) and Atari RAM of OpenAI Gym (Brockman et al., 2016), where each benchmark is allowed less than 100 expert state-action pairs. The proposed method outperforms the previous AIL methods by a significant margin in scenarios with a small number of perfect or imperfect expert demonstrations. We discover that the proposed corruption method generates various transformed samples that are not out-of-distribution. We perform comprehensive ablation studies, detailing the intuitions and effects of various design choices and factors. Through runs with various hyperparameters on non-image control tasks, we demonstrate that the corruption rate and loss function should be carefully chosen.

2. Related Work

2.1 Data-efficient Reinforcement Learning

In deep RL, studies have been conducted to improve sample efficiency. For continuous control, several studies have suggested the use of reconstruction loss (Lee et al., 2020; Hafner et al., 2019). However, most of the suggested methods are RL benchmarks, which have a sparse reward or image state. Methods using a self-supervised error as an intrinsic reward have been proposed to improve the sample efficiency in a sparse reward scenario (Pathak et al., 2017; Mohamed and Jimenez Rezende, 2015; Simmons-Edler et al., 2019; Wu et al., 2022). For the image state, various image augmentation techniques and self-supervised objectives have been applied to reduce environmental interactions (Srinivas et al., 2020; Schwarzer et al., 2020; Kielak, 2020; Yarats et al., 2020; Laskin et al., 2020; Mazoure et al., 2020; Hansen et al., 2020, 2021; Zadaianchuk et al., 2022). Additionally, there are some works that apply self-supervised learning to train latent space or feature embedding for reward learning (Brown et al., 2020; Bobu et al., 2023). Different from most existing approaches, we utilize SSL to improve the sample efficiency of expert demonstrations for non-image imitation learning.

2.2 Self-supervised Representation Learning

Currently, SSL is divided into three approaches. The first is a pretext task (Noroozi and Favaro, 2016), which creates a pre-task that can learn useful feature representations and use the learned representations on downstream tasks. The second is UCL. Contrastive learning works on a simple push-pull principle, and it can be a sample or cluster level (Caron et al., 2020). The contrastive loss contrasts the neighboring instances with non-neighboring ones (Chen et al., 2020b; He et al., 2020). The third is asymmetric twin-based SSL. Unlike UCL,

these methods do not use negative samples during training. Asymmetric twin-based SSL methods learn robust representations in such a way that differently transformed versions of input have the same representation. As representative methods, BYOL (Grill et al., 2020) and Simsiam (Chen and He, 2021) used Siamese networks with weight sharing and stop-gradient techniques to avoid collapse. Barlow twins (Zbontar et al., 2021) utilized a correlation matrix between the representations of a differently transformed same input to maximize the similarity between them while minimizing redundancy. VICReg (Bardes et al., 2021) is effective for making the two representations similar and reducing the embedding of non-informative factors.

There are two SSL methods for tabular data, VIME (Yoon et al., 2020) and SCARF (Bahri et al., 2021), which can be directly applied to non-image control benchmarks. VIME uses a random corruption method and SCARF suggests a method that replaces each feature dimension by a random draw from that feature dimension's empirical marginal distribution. The difference between our work is the corrupted data of swapping is a mixture of only two state-action pairs because the replaced features are sampled from another single state-action pair. Meanwhile, the corrupted data of SCARF is a mixture of varying state-action pairs, resulting in possible out-of-distribution data. In the experiments, the proposed swapping corruption method showed higher performance compared to the other two methods. We measured the variance and outlier scores of the corrupted samples produced by VIME, SCARF, and the proposed method, to confirm that the proposed method qualitatively generates more realistic and still varied samples.

2.3 Inverse Reinforcement Learning

Although IRL (Abbeel and Ng, 2004; Ziebart et al., 2008a) has made significant advances in IL, it encounters some challenges. First, there is an unstable training issue for adversarial learning; improved algorithms have been proposed to overcome this problem. GAIL (Ho and Ermon, 2016) is the first study drawing an analogy between IL and generative adversarial networks (Goodfellow et al., 2014). AIRL (Fu et al., 2018) proposed an AIL method that is robust to changes in dynamics. VAIL (Peng et al., 2019) improves the stability problem by constraining the information flow in the discriminator. EAIRL (Qureshi et al., 2018) reduces the overfitting problem using empowerment (the information gain on action entropies). These algorithms have been suggested to improve the stability and scalability of AIL.

Second, studies on the sample efficiency of expert demonstrations have not been sufficiently conducted. SAILfO (Torabi et al., 2019) covers the necessity of studying the sample efficiency of expert demonstrations, and proposes a simple model-based algorithm. f-GAIL (Zhang et al., 2020) showed that finding an appropriate discrepancy measure during training is better than using a predetermined measure to improve sample efficiency. ASAF (Barde et al., 2020) is an algorithm in which training the discriminator could perform the role of policy and showed that it helps improve sample efficiency. However, these methods require at least five full trajectories to recover the expert policy on continuous control benchmarks such as the MuJoCo physics engine. Unlike those methods, the proposed method successfully imitates the expert's behavior with less than one full trajectory.

In practice, it is difficult to collect perfectly optimal demonstrations because demonstrations are commonly collected by crowdsourcing (Hu et al., 2018) or multiple experts.

The collected data from external sources are normally imperfect—a mixture of optimal and non-optimal demonstrations. To address these problems, algorithms for IL from imperfect demonstrations have been proposed (Wu et al., 2019; Zhang et al., 2021). We demonstrate that combining the proposed method with other algorithms for IL from imperfect demonstrations further improves them, thus, verifying the scalability of the proposed method.

3. Theoretical Motivation

This section aims to provide the theoretical intuition for designing our cost function and feature space aimed at reducing the generalization gap.

In a general classification learning process, the data and label spaces are denoted by $\mathcal{X}$ and $\mathcal{Y}$, respectively, and $\mathcal{P}(x, y)$ is the joint distribution of the data and label. The primary objective is to learn a classifier $f : \mathcal{X} \to \mathcal{Y}$ that minimizes the expected value of a loss term ℓ over the joint distribution $\mathcal{P}(x, y)$. The classifier f is composed of a feature extractor h and a fully connected layer $\mathbf{W}$. More formally, when $\mathcal{P}(x, y)$ is known, the expected risk can be defined as follows:

$$R(f|\mathcal{P}) = \int \ell(\mathbf{W}^\top h(x), y) d\mathcal{P}(x, y). \tag{1}$$

However, in practice, the data distribution is unknown. Empirical risk minimization (ERM) is commonly utilized to address this problem. Then, the minimization can be defined as follows:

$$\hat{R}(f|\mathcal{P}) = \frac{1}{M} \sum_{i=1}^{M} \ell(\mathbf{W}^\top h(x_i), y_i), \tag{2}$$

where M is the number of training data points. The generalization performance of a practical classification learning process is highly dependent on the volume of training data. However, in many real-world scenarios, acquiring sufficient training data is a challenging and time-consuming task.

The generalization of ERM has been theoretically justified based on Vapnik-Chervonenkis (VC) theory (Vapnik, 1999). The upper bound of the expected risk of the classifier f is formulated by

$$R(f|\mathcal{P}) \leqslant \hat{R}(f|\mathcal{P}) + O\left(\left(\frac{|\mathcal{F}|_{\mathrm{VC}} - \log \delta}{M} \right)^{\xi} \right), \tag{3}$$

where $0 \leq \delta \leq 1$, $0.5 \leq \xi \leq 1$, and $|\mathcal{F}|_{\mathrm{VC}}$ is the finite VC dimension of a classifier.

Let $\mathcal{H} \subseteq \mathbb{R}^D$, where D represents latent dimension, be the feature space of h and $\mathbf{W} \in \mathbb{R}^{D \times C}$, where C denotes the number of classes, be the last fully connected layer of f.

Assumption 1. *Consider a perfectly trained fully connected layer $\mathbf{W}^*$. We can then split the features as class-relevant or class-irrelevant based on their mutual information with a class label Y. Class-relevant features have high mutual information between $h_d(x)$ for $1 \leq d \leq D$ and Y, while class-irrelevant features have low mutual information between $h_d(x)$ for $1 \leq d \leq D$ and Y.*

Definition 1. *(He et al., 2019) To minimize a change in classification results by variation of class-irrelevant features, $w_{d,i}$ and $w_{d,j}$ for $1 \leq d \leq D$ and $1 \leq i, j \leq C$ should*

be similar for class-irrelevant features in the empirical risk on original data $\hat{R}(f|\mathcal{P}) = \frac{1}{M} \sum_{i=1}^{M} \ell(\mathbf{W}^\top h(x_i), y_i)$.

Definition 1 from He et al. shows that in $\hat{R}(f|\mathcal{P})$, $w_{d,i}$ for all i for class-irrelevant features are not forced to be 0, that is, f^* preserves class-irrelevant features.

Eq. 3 highlights three key factors that can improve the generalization performance of a classifier: 1) minimizing the empirical risk $\hat{R}(f|\mathcal{P})$, 2) increasing the size of the training data, and 3) reducing the VC dimension of the classifier. To improve the generalization, we propose an approach that leverages self-supervised learning. By utilizing transformed data as its own supervision, self-supervised learning can learn feature representations that filter out information about nuisance variables. This enables the model to capture more meaningful and generalizable features that can improve the overall performance of the classifier.

To begin with, we aim to reduce the upper bound of the expected risk by minimizing the discrepancy between the original and transformed samples in the feature space $\mathcal{H}$. Formally, we utilize the mean squared error (MSE), the most widely-used discrepancy loss, as follows:

$$\mathcal{L} = \frac{1}{M \times G} \sum_{i=1}^{M \times G} \left\| h(x_i) - h(x_i') \right\|_2^2, \tag{4}$$

where x' represents the transformed version of x and $M \times G$ is the number of training data with a finite constant number of augmentations G.

By minimizing Eq. 4, we can effectively learn a feature space that does not encode the distortions present in the input samples due to transformations. In other words, the resulting feature representation is invariant to such distortions, leading to improved generalization performance. Given the feature extractor with Eq. 4 as h_{MSE} and the corresponding feature space as $\mathcal{H}_{\mathrm{MSE}}$, the classifier $f_{h_{\mathrm{MSE}}}$ consisting of h_{MSE} holds the following empirical risk equation:

$$R(f_{h_{\mathrm{MSE}}}|\mathcal{P}) \leqslant \hat{R}(f_{h_{\mathrm{MSE}}}|\mathcal{P}) + O\left(\left(\frac{|\mathcal{F}_{h_{\mathrm{MSE}}}|_{\mathrm{VC}} - \log \delta}{M}\right)^\xi\right), \tag{5}$$

where $|\mathcal{F}_{h_{\mathrm{MSE}}}|_{\mathrm{VC}}$ is the finite VC dimension of the classifier with the feature space $\mathcal{H}_{\mathrm{MSE}}$ trained with Eq. 4 simultaneously.

Corollary 1. *h_{MSE} is effective in reducing the VC dimension, however, the problem is that the upper bound of $R(f|\mathcal{P})$ we seek to is based on $\hat{R}(f|\mathcal{P})$ instead of $\hat{R}(f_{h_{MSE}}|\mathcal{P})$. As a result, $f_{h_{MSE}}$ cannot capture all the properties preserved in $\hat{R}(f|\mathcal{P})$. Additionally, due to the distribution gap between the feature spaces h and h_{MSE}, the optimal function $f_{h_{MSE}}^*$ of $\hat{R}(f_{h_{MSE}}|\mathcal{P})$ is not guaranteed to be a minimum of $\hat{R}(f|\mathcal{P})$.*

Proof. The proof can be found in Appendix A.1.

The feature extractor f_h typically preserves both class-relevant and class-irrelevant features, whereas the embedding learned by $f_{h_{\mathrm{MSE}}}$ is more heavily regularized due to the intensive transformation applied to the input samples. This regularization can be beneficial for reducing the upper bound of the expected risk of the classifier, but it can also introduce a distribution shift. To mitigate this issue, unsupervised contrastive learning can be utilized. We assume that MSE and contrastive learning share the feature space.
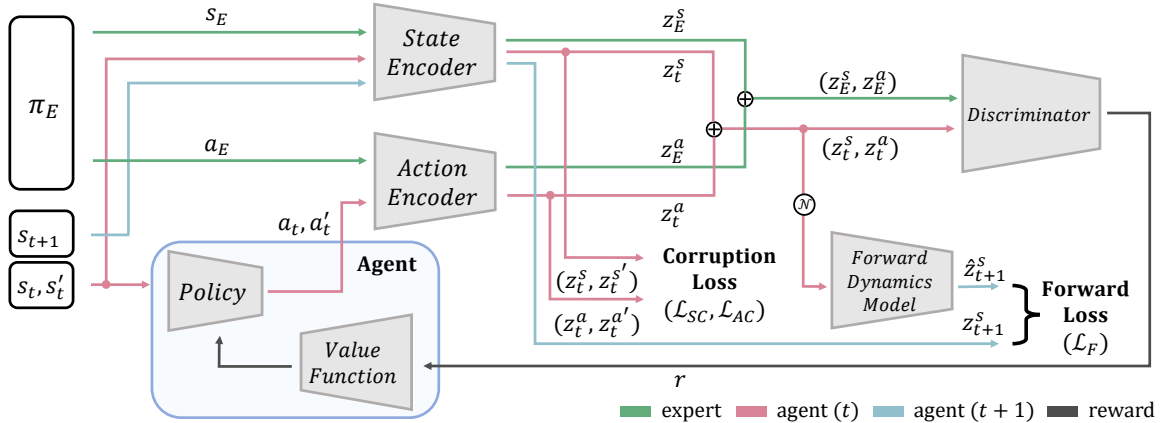
Figure 1: Overview of the proposed model. Our proposed model comprises six networks during IRL training. 1) The policy generates actions a based on states s according to a specified policy; 2) The value function evaluates the current policy π_θ. It is trained using rewards r derived from an estimated cost function (discriminator); 3) The state encoder extracts a feature representation of raw states s, 4) The action encoder extracts a feature representation of actions a, 5) The forward dynamics model predicts the feature representation of the distorted version of next states $\hat{z}^s_{t+1}$ based on the feature representations of the current state and action, z^s_t and z^a_t, along with Gaussian noises $\mathcal{N}$, and 6) The discriminator discriminates agent demonstrations from expert demonstrations. The input is $z^s \oplus z^a$. The discriminator is also referred to as a cost function. More details of each component and loss in the figure are described in Section 4.

Definition 2. *(Wang and Isola, 2020) Consider a perfectly trained* $\mathbf{W}^*$ *via contrastive learning. The feature space* $\mathcal{H}_{MSE}$ *becomes maximally informative.*

Claim 1. *Class-irrelevant yet informative features are preserved via contrastive learning. It reduces the inconsistency between* $\hat{R}(f_{h_{MSE}}|\mathcal{P})$ *and* $\hat{R}(f|\mathcal{P})$.

Proof. The proof with assumptions can be found in Appendix A.2.

By combining MSE (Eq. 4) with contrastive learning in the feature space, we anticipate a notable enhancement in generalization, even when dealing with a modest amount of training data. Our suggested algorithm expands this theoretical foundation rooted in classification to AIL. Importantly, we capture temporally significant features crucial in RL by creating a transformed version based on temporal features for contrastive learning.

4. Method

The core of RL is an agent and environment. An agent receives a reward from the environment based on the actions determined by a policy. RL learns the optimal policy of a Markov decision process. For IL, the agent learns the optimal policy where a pre-defined reward function does not exist by relying only on given expert demonstrations. We define the IL scenario from a small number of expert demonstrations. The expert demonstrations $\mathcal{D}_E$ are

sampled from a state-action density of an expert, ρ_O, defined as follows:

$$\mathcal{D}_E = \{(s_n, a_n)\}_{n=1}^{N_E} \overset{i.i.d.}{\sim} \rho_O(s, a), \tag{6}$$

where N_E is the number of state-action pairs from ρ_O. We assume a scenario in which N_E is less than the number of one full trajectory. We denote a state-action pair by $x = (s, a)$ where $x \in X$ and $X = S \times A$.

This study comprises the following six networks, as illustrated in Figure 1.

- **A policy** $\pi_\theta(\cdot)$ parameterized by θ that generates actions a given states s based on a policy.

- **A value function** $V(\cdot)$ that evaluates a current policy π_θ. V is trained with rewards r from an estimated cost function.

- **A state encoder** $SE(\cdot)$ that extracts a feature representation of raw states s. $z^s = SE(s) \in Z^s$.

- **An action encoder** $AE(\cdot)$ that extracts a feature representation of actions a. $z^a = AE(a) \in Z^a$.

- **A forward dynamics model** $F(\cdot)$ that predicts the feature representation of the distorted version of next states $\hat{z}_{t+1}^s$ from the feature representations of the current state and action, z_t^s and z_t^a, and Gaussian noises $\mathcal{N}$. $\hat{z}_{t+1}^s = F(z_t^s \oplus z_t^a \oplus \mathcal{N}) \in \hat{Z}_{t+1}$, where $\oplus$ is concatenation.

- **A discriminator** $D_\omega(\cdot)$ parameterized by ω that discriminates agent demonstrations from expert demonstrations $\mathcal{D}_E$. The input of D_ω is $z^s \oplus z^a$. D_ω is also called a cost function. In the RL phase, D_ω is the estimated cost function ($r = -\log(D_\omega(z^s \oplus z^a)) \in R$).

As shown in Algorithm 1, the proposed method comprises three major parts. In Section 4.1, we explain how to train the cost function D_ω using expert demonstrations $\mathcal{D}_E$ (GAIL in Algorithm 1). In Section 4.2, we describe how to use SSL in a non-image environment (REPR in Algorithm 1). We implement trust region policy optimization (Schulman et al., 2015) to train the agent policy π_θ by following the use in Ho and Ermon (2016) (TRPO in Algorithm 1).

4.1 Generative Adversarial Imitation Learning

Our method is based on generative adversarial imitation learning (GAIL) (Ho and Ermon, 2016). GAIL finds an optimal policy by matching an occupancy measure between expert E and the agent. The optimization equation of GAIL can be derived in the form of the Jensen-Shannon divergence, which is equal to the minimax equation of generative adversarial networks (Goodfellow et al., 2014). The minimax optimization of GAIL is expressed as follows:

$$\min_\theta \max_\omega \underset{x \sim \mathcal{D}_\pi}{\mathbb{E}} \left[\log D_\omega(x)\right] + \underset{x \sim \mathcal{D}_E}{\mathbb{E}} \left[\log(1 - D_\omega(x))\right], \tag{7}$$

Algorithm 1 Sample-efficient Adversarial Imitation Learning

1: **input**: Expert demonstrations $\mathcal{D}_E \triangleq \{x_i\}_{i=1}^{N_E}$, # of batches B, Training epochs T.
2: **for** $k \leftarrow 1$ to T **do**   ▷ Section 4
3:     Obtain trajectories $\mathcal{D}_k = \{x_{k,i}\}_{i=1}^N$ using π_θ
4:     $\pi_\theta \leftarrow \text{TRPO}(\pi_\theta, V, D_\omega, \mathcal{D}_k)$ ▷ Ho and Ermon (2016)
5:     $SE, AE \leftarrow \text{REPR}(SE, AE, F, \mathcal{D}_k)$
6:     $D_\omega \leftarrow \text{GAIL}(D_\omega, SE, AE, \mathcal{D}_k, \mathcal{D}_E)$
7: **end for**
8: **function** REPR($SE, AE, F, \mathcal{D}_k$)   ▷ Section 4.2
9:     **for** $b \leftarrow 1$ to B **do**
10:         Generate X_b' by Eq. 11   ▷ Section 4.2.2 Corruption Method
11:         Obtain Z_b from $\mathcal{D}_{k,b}$ using (SE, AE)
12:         Obtain Z_b' from X_b' using (SE, AE)
13:         Update SE, AE, and F by Eq. 14
14:     **end for**
15:     **return** SE, AE ▷ Learning state and action representations
16: **end function**
17: **function** GAIL($D_\omega, SE, AE, \mathcal{D}_k, \mathcal{D}_E$)   ▷ Section 4.1
18:     **for** $b \leftarrow 1$ to B **do**
19:         Obtain Z_b from $\mathcal{D}_{k,b}$ using (SE, AE)
20:         Update SE, AE, and D_ω by Eq. 8
21:     **end for**
22:     **return** D_ω ▷ Learning cost function
23: **end function**

---

where $\mathcal{D}_\pi$ and $\mathcal{D}_E$ are the corresponding demonstrations from an agent policy π_θ and expert policy π_E, respectively. In GAIL, the raw state and action are input to the discriminator. For the proposed discriminator, state and action representations embedded by state and action encoders, $SE(\cdot)$ and $AE(\cdot)$, are input. The discriminator is expressed as follows:

$$\max_\omega \underset{x \sim \mathcal{D}_\pi}{\mathbb{E}} [\log D_\omega(z)] + \underset{x \sim \mathcal{D}_E}{\mathbb{E}} [\log(1 - D_\omega(z))], \tag{8}$$

where $z = z^s \oplus z^a$. Here, z^s is a state representation embedded by $SE(s)$, and z^a is an action representation embedded by $AE(a)$.

4.2 State and Action Representations

4.2.1 MODELING FORWARD DYNAMICS

Each domain requires different ways of generating self-supervision depending on the properties of the data. For example, BERT (Devlin et al., 2018) leverages a training signal by predicting future words from previous words. For RL, the prediction error of a forward dynamics model has been used as an intrinsic reward (Pathak et al., 2017; Schwarzer et al., 2020). In tabular data, in contrast to data from an image, it is difficult to create a distorted version of an original input without losing semantic information. Therefore, we posit that maximizing the agreement between the distorted and original ones is more suitable for learning meaningful

features than maximizing the agreement between the two distorted versions of input in tabular data. We propose a method that generates a distorted version of the input to learn or discard the desired features and their corresponding loss function for RL.

The proposed method uses the forward dynamics model to predict the distorted version of the next state representation from the given current state and action representations. First, the forward dynamics model is mathematically expressed as follows:

$$\hat{z}_{t+1}^s = F(z_t^s \oplus z_t^a \oplus \mathcal{N}), \tag{9}$$

where $z_t^s = SE(s_t)$, $z_t^a = AE(a_t)$ at a time step t, and $\mathcal{N}(0,1)$ is the Gaussian noise. The output $\hat{z}_{t+1}^s$ represents a distorted version of the observed future representations z^{t+1}. The choice of a transformation controls what the representation learns. Thus, we apply a distortion by concatenating Gaussian noise rather than using a corrupted state-action pair because corruption cannot guarantee consistency in information with respect to temporality.

We use a contrastive loss for training. InfoNCE-based unsupervised contrastive learning (UCL) methods learn a feature representation by maximizing the agreement between differently transformed same input while minimizing that of the rest of the input (negative samples). The learned representation from UCL is invariant in unnecessary details; however, it contains maximal information by maximizing a lower bound on the mutual information between the two views (Wang and Isola, 2020; Poole et al., 2019). We utilize the InfoNCE loss to obtain as many temporally informative features as possible. The proposed forward dynamics model is trained to maximize the agreement between the distorted and observed next state representations while minimizing that of the rest of the state representations. This is expressed as follows:

$$\mathcal{L}_F = -\mathbb{E}\left[\log\frac{e^{\text{cs}(\hat{z}_{i,t+1}^s, z_{i,t+1}^s)/\tau}}{\zeta}\right], \quad \zeta = \sum_{j=1}^{BS}\mathbb{1}_{j\neq i}e^{\text{cs}(\hat{z}_{i,t+1}^s, z_{j,t}^s)/\tau} + \sum_{j=1}^{BS}\mathbb{1}_{j\neq i}e^{\text{cs}(\hat{z}_{i,t+1}^s, z_{j,t+1}^s)/\tau},$$
$$\tag{10}$$

where $(z_t^s, z_{t+1}^s, \hat{z}_{t+1}^s) \sim (Z_t^s, Z_{t+1}^s, \hat{Z}_{t+1}^s)$ and $\text{cs}(u,v) = u^\top v/||u||\,||v||$ (j indexes the state or next state representation in the batch, and BS is the batch size). Following SimCLR (Chen et al., 2020b), we use the other $2(BS-1)$ representations in the batch as negative samples.

4.2.2 Corruption Method

Learning representations that can discard nuisance features is preferable to reduce sample complexity. To help with this, we propose a corruption method that creates a distorted sample showing diverse views that are possible in-distribution. The proposed method swaps the input features of each state-action pair with the input features of the same indices of another state-action pair in a batch. For a batch of state-action pairs sampled from the current policy, X_b, we generate a corrupted version x_i' for each state-action pair x_i. The corrupted versions of state s and action a are generated, respectively. However, for convenience of rationalization, we explain the method based on the state-action pair x.

First, we make a copy of X_b as X_b^c and permute X_b^c by changing the order of each state-action pair in the batch at random, $perm(X_b^c)$. Second, we sample some indices of the state-action pair without replacement, $I \in \{0, ..., dim(s \oplus a) - 1\}^q$. q is the number of features to corrupt $(= \lfloor c \cdot dim(s \oplus a) \rfloor)$, where c is a corruption rate $(c = c^s + c^a)$. Third,

we duplicate I as a shape of $N(X_b) \times q$. Subsequently, we generate corrupted state-action pairs X' of a given batch as follows:

$$
\begin{aligned}
X_b^c[I] &= perm(X_b^c)[I], \\
X_b' &= X_b^c,
\end{aligned}
\tag{11}
$$

where $X_b' = \langle S_b' \times A_b' \rangle$. For convenience, we omit subscripts b hereafter. We refer to this method as the swapping corruption method. Empirically, we observed superior performance compared to existing methods on non-image control benchmarks.

Numerous discrepancy measures can quantify the similarity between corrupted and original inputs. For the state, we maximize the similarity between the representation of the distorted and observed versions by minimizing the mean squared error (MSE) as follows:

$$
\mathcal{L}_{SC} = \mathbb{E} \left\| z^s - z^{s'} \right\|_2^2,
\tag{12}
$$

where $(z^s, z^{s'}) \sim (Z^s, Z^{s'})$ and $Z^s = SE(S)$ and $Z^{s'} = SE(S')$. For the state representation, temporally predictive features should be embedded well to minimize $\mathcal{L}_F$ simultaneously. Thus, we observed the best performance with MSE compared to other indirect discrepancy measures, as shown in Tab 7. For the action representation, because the gradients from the $\mathcal{L}_F$ are not sufficient to hinder collapse, we use the Barlow twins loss (Zbontar et al., 2021) that is robust against the constant embedding problem. It has been proven through connection with mutual information that the Barlow twins loss also learns the feature representation which is invariant to the distortion of the sample (Zbontar et al., 2021), similar to the MSE. The Barlow twins loss draws the cross-correlation matrix close to the identity matrix. This is expressed as follows:

$$
\mathcal{L}_{AC} = \sum_i \left(1 - \mathcal{C}_{ii}\right)^2 - \sum_i \sum_{j \neq i} \mathcal{C}_{ij}{}^2, \qquad \mathcal{C}_{ij} = \frac{Z_i^a Z_j^{a'}}{\sqrt{(Z_i^a)^2}\sqrt{(Z_i^{a'})^2}}
\tag{13}
$$

where $Z^a = AE(A)$ and $Z^{a'} = AE(A')$. i, j index the vector dimensions of the action representations. Therefore, the Barlow twins loss prevents collapse by maximizing the similarity between the representation of the distorted and observed versions of action and reducing entanglement between the components of the representations. Ablations that can give more intuitions about losses are given in Section 6. Consequently, the total loss for SSL is computed as follows.

$$
\mathcal{L}_{SS} = \lambda_F \mathcal{L}_F + \lambda_S \mathcal{L}_{SC} + \lambda_A \mathcal{L}_{AC},
\tag{14}
$$

where λ_F, λ_S, and λ_A are hyperparameters for each loss.

5. Experiments

The efficacy of the proposed approach is assessed using MuJoCo (Todorov et al., 2012) and Atari RAM of OpenAI Gym (Brockman et al., 2016), with each benchmark having fewer than 100 expert state-action pairs. In scenarios with a limited number of perfect or imperfect expert demonstrations, our method surpasses previous Adversarial Imitation Learning (AIL) approaches by a significant margin. We assessed the performance of the proposed method on

Table 1: Final performance using 100 expert state-action pairs on Ant-v2, HalfCheetah-v2, and Walker2d-v2 of MuJoCo. Best results are in **bold**. The proposed method outperforms existing IRL methods by a significant margin. It succeeds at imitating the expert's behavior on all three benchmarks using only 100 expert state-action pairs.

	BC	GAIL	AIRL	VAIL	EAIRL	SQIL	ASAF	Ours
Ant	932.2±171.7	4198.2±72.6	3922.9±210.7	4216.8±31.0	3137.5±424.8	-141.4±427.6	1015.6±107.0	**4554.8±162.6**
HalfCheetah	1875.2±1623.3	2034.6±2384.6	-214.1±45.2	-1012.8±497.1	6.6±15.0	-238.0±22.5	1187.6±1935.9	**5416.0±203.8**
Walker2d	535.5±134.4	3513.4±172.9	909.7±695.8	3466.7±109.0	2084.9±2499.7	283.3±26.5	192.9±58.5	**3527.6±131.4**
Average	1114.3±688.1	3248.8±876.7	1539.5±317.2	2223.6±212.4	1743.0±979.8	-32.0±158.9	798.7±700.4	**4499.4±165.9**

Table 2: Final performance when using the proposed corruption method (Swapping) and existing methods ($N_E = 100$), and variance and predicted local outliers (Breunig et al., 2000) of corrupted states. For measuring the outlier factor of corrupted states, we make use of 10 neighbors from observed states.

Ant	Random	Mean	Each dim	Swapping
Cumulative rewards	4263.7±243.9	4482.0±127.4	4459.3±128.4	**4554.8±162.6**
Variance ↑	0.765±0.05	0.756±0.03	**0.843±0.02**	**0.843±0.02**
Predicted local outliers (%) ↓	90%±8%	**6%±5%**	26%±8%	11%±6%

five continuous control benchmarks simulated by MuJoCo (Ant-v2, HalfCheetah-v2, Hopper-v2, Swimmer-v2, and Walker2d-v2) in four distinct settings: using expert demonstrations of less than one full trajectory with the optimality of 25%, 50%, 75%, or 100%. We tested the sample efficiency of the proposed method in a scenario where optimal demonstration samples of less than one full trajectory are available (≤ 100). Expert demonstrations with optimalities of 25%, 50%, and 75% represent imperfect demonstrations - a mixture of optimal and non-optimal demonstrations. Imperfect demonstrations $\mathcal{D}_I$ are sampled from a noisy state-action density ρ, expressed as follows: $\mathcal{D}_I = \{(s_n, a_n)\}_{n=1}^{N_I} \overset{i.i.d.}{\sim} \rho(s, a)$, where N_I is the number of state-action pairs from ρ. The noisy state-action density ρ can be expressed as follows:

$$\rho(s, a) = \psi\rho_O(s, a) + \sum_{i=1}^{n} v_i\rho_i(s, a)$$
$$= \psi\rho_O(s, a) + (1 - \psi)\rho_N(s, a),$$

(15)

where ρ_O is the state-action density of an expert, ρ_i is the state-action density of a single non-expert, and n is the number of non-experts. Furthermore, ψ, satisfying $0 < \psi < 1$, is an unknown mixing coefficient of the optimal and non-optimal state-action densities, and $\psi + \sum_{i=1}^{n} v_i = 1$; that is, an optimality of 25% indicates that $\psi = 0.25$. More details on the experimental setting, definition, and hyperparameters can be found in Appendix B.

Optimality of 100% First, we evaluated the proposed method with a small number of perfect expert demonstrations. We compared the method with seven existing IL methods: BC, GAIL, AIRL, VAIL, EAIRL, SQIL, and ASAF. Table 1 shows that the proposed method outperforms other IL methods on all three benchmarks. Particularly, the proposed method succeeded in perfectly imitating the expert policy on HalfCheetah. Notably, GAIL and VAIL show higher performance than the recently proposed ASAF in 100 state-action pairs. For f-GAIL, we conducted experiments using the official GitHub; however, it failed to converge in

less than one full trajectory. We surmise that this is because a reasonable number of expert state-action pairs must be guaranteed to automatically learn an appropriate discrepancy measure for the given pairs.

We also tested our method with varying expert data sizes. As provided in Table 12 in Appendix D, there is a relatively small decrease in the performance up to $N_E = 20$ on all three benchmarks. However, when $N_E = 10$, the performance is decreased by a large margin. To make the experimental results stronger, we tested the reliability of the reported average using IQM (Agarwal et al., 2021). We obtained IQM 4555.8 on Ant, IQM 5420.3 on HalfCheetah, and IQM 3527.9 on Walker2d, which are very close to the reported average.

In addition, we tested the proposed method (without $\mathcal{L}_{AC}$) on two discrete control benchmarks of OpenAI Gym (Brockman et al., 2016; BeamRider-ram-v0, and SpaceInvaders-ram-v0). We observed that the cumulative rewards of the proposed method are superior to those of the GAIL. For the results on the two discrete control benchmarks, please refer to Appendix F. Moreover, the average cumulative rewards of the expert policy that we obtained can be found in Appendix B.1.

Corruption method   We showed the performance of the swapping corruption method by comparing it with the existing two corruption methods and additionally, a mean corruption method, which replaces the features with the empirical marginal distribution's mean, in cumulative rewards, variance, and local outlier score. Table 2 shows that the proposed method shows a higher performance compared with the three corruption methods. We observed that the proposed method generated transformed samples that provide more diverse views compared with random and mean methods and comparably diverse views compared with the method, obtaining each replaced feature from varying state-action pairs (Each dim). We measured its diversity on the corrupted states using variance σ^2 : $\frac{1}{1000} \sum_{i=1}^{1000} \left(\sum_{j=1}^{dim(s)} (s'_{i,j} - \bar{s}'_{i,j})^2 \right)$ where $\bar{s}'$ denotes the average of $\{s'_i\}_{i=1}^{1000}$.

Also, one potential concern about using corruption as a transformation technique is that the corrupted samples are out-of-distribution, resulting in performance degradation. To evaluate this, we computed the local outlier factor (Breunig et al., 2000) of the corrupted states. We computed the percentage of corrupted samples that are local outliers with respect to the observed states. Table 2 shows that for the random method, the most corrupted states are predicted as outliers, despite the low variance. For the mean method, the corrupted states are mostly realistic; however, the diversity is lower compared with the swapping method, which limits the effectiveness of data augmentation. Replacing each feature with a feature from a different combination has diversity, but it also creates more out-of-distribution data. The proposed swapping method showed high variance and a relatively low percentage of local outliers. As a result, we empirically confirmed that for control tasks, it can be beneficial to create meaningful in-distribution data in the corruption process.

Optimality of 25%, 50%, or 75%   We tested the performance of the proposed method in a more practical scenario with imperfect demonstrations. In this environment, we combined the proposed method with 2IWIL (Wu et al., 2019). 2IWIL showed stable training compared to other IL algorithms for imperfect demonstrations because they predict the confidence of given demonstrations in a pre-stage. Please refer to Appendix B.2 for detailed explanations about 2IWIL, other comparison methods, and the collected imperfect demonstrations. The pseudo-code of the combined algorithm can be found in Appendix C.
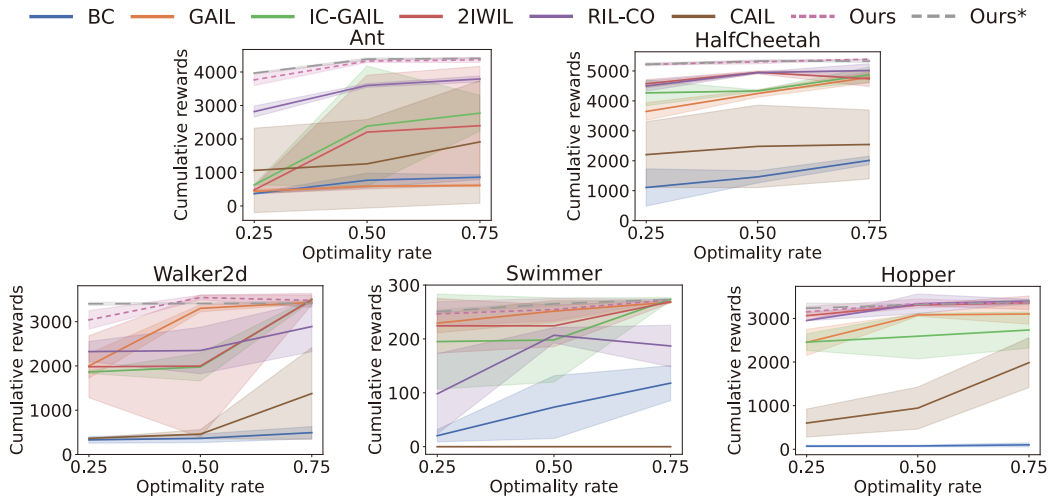
Figure 2: Final performance on five continuous control benchmarks with different optimality rates ψ. Vertical axes denote cumulative rewards acquired during the last 1000 training iterations. Shaded regions denote standard errors over three runs. Ours* = Ours + MM

Table 3: Final performance using 25 optimal and 75 non-optimal state-action pairs ($\psi = 0.25$) to test improvement in sample efficiency.

Ablation		Cumulative Rewards		
MM	Ours	Ant	HalfCheetah	Walker2d
		478.4 ± 159.7	4573.1 ± 194.4	1981.8 ± 989.7
✓		776.3 ± 1805.1	4728.3 ± 172.8	$\mathbf{3436.0\pm15.8}$
	✓	3764.6 ± 241.2	5217.4 ± 29.7	3039.6 ± 295.6
✓	✓	$\mathbf{3966.0\pm26.2}$	$\mathbf{5221.0\pm75.1}$	3403.0 ± 52.5

Figure 2 shows the cumulative rewards on the five continuous control MuJoCo benchmarks with different optimality rates. The proposed method combined with 2IWIL outperforms the other six comparisons by a large margin in all optimality rates. Particularly, the relative improvement over 2IWIL and CAIL (Zhang et al., 2021) on Ant is 288% and 208% on average, respectively. However, we observed a decrease in the cumulative rewards when the noise rate was 0.75 on Ant and Walker2d benchmarks. We surmise that this was caused by a deficiency in the number of locomotion movements from the optimal policy. The degree of improvement in performance is dependent on a number of given optimal demonstrations to some extent because SSL methods create an auxiliary training signal by leveraging the given data.

Manifold mixup We applied a widely-used sample efficiency technique, manifold mixup (MM) (Verma et al., 2019b), to the combined method. In previous studies (Mangla et al., 2020; Lee et al., 2021), it is shown that MM in the feature space enriched by SSL is further effective to improve performance. Many mixup-based augmentation studies have been proposed to improve the generalization of image domains (Yun et al., 2019; Hendrycks et al., 2019; Kim et al., 2020; Dabouei et al., 2021) and sequential domains (Chen et al., 2020a; Guo et al., 2019). However, for RL, the application of Mixup is only applicable to image environments (Wang et al., 2020; Singh et al., 2019). Through this experiment, we would like

Table 4: Ablation studies using 100 expert state-action pairs to test the importance of $\mathcal{L}_F$ and $(\mathcal{L}_{SC}, \mathcal{L}_{AC})$. FD = Forward dynamics, and CR = Corruption.

Ablation		Cumulative Rewards		
$\mathcal{L}_F$	$\mathcal{L}_{SC}, \mathcal{L}_{AC}$	Ant	HalfCheetah	Walker2d
		4198.2±72.5	2034.6±2384.6	3513.4±172.9
✓		3329.6±513.0	2330.3 ±3741.3	3524.5±17.3
	✓	2244.0±208.6	591.9±549.9	1028.6±15.4
✓	✓	**4554.7±162.5**	**5415.9±203.8**	**3527.5±131.3**

Table 5: Ablation studies using 50 optimal and 50 non-optimal state-action pairs to test sensitivity to corruption rate of state and action.

c^a			0.2			0.5	$<= 0.5$
c^s	0.1	0.2	0.3	0.4	$<= 0.5$	$<= 0.5$	$<= 0.5$
Ant	**4384.71±49.21**	4282.89±170.53	4206.76±502.76	4127.30±451.87	3546.28±487.71	2425.96±65.69	560.64±326.97

to compare the efficiency of 1) MM, 2) the proposed method, and 3) using both as a sample efficiency technique. MM increases the diversity of expert demonstrations by interpolating the feature space output of the input data pair. We performed MM on the feature space as follows: $(\bar{z}, \bar{y}) = (\text{Mix}_\lambda(z_i, z_j), \text{Mix}_\lambda(y_i, y_j))$, where $\text{Mix}_\lambda(a, b) = \lambda \cdot a + (1 - \lambda) \cdot b$. Here, (z_i, z_j) are the feature representations of (x_i, x_j), and (y_i, y_j) are the estimated confidence by 2IWIL. Table 3 shows that, when only MM was used, the performance on Walker2d improves; however, the performance on Ant and HalfCheetah does not show a reasonable improvement. This indicates that it is difficult to naturally learn representations that are temporally predictive and robust to diverse distortions from training signals generated by synthetic data from MM. Conversely, when only the proposed method was used, we observed a relatively small increase on Walker2d due to a deficiency in the number of optimal locomotion movements. Consequently, as shown in Figure 2 and Table 3, we observed near-optimal performance on all the benchmarks with varying optimalities when both ours and MM were used.

6. Building Intuitions with Ablations

We conducted ablations on the proposed factors to provide an intuition of each role.

Importance of both $\mathcal{L}_F$ and $(\mathcal{L}_{SC}, \mathcal{L}_{AC})$ Table 4 shows that $\mathcal{L}_F$ and $(\mathcal{L}_{SC}, \mathcal{L}_{AC})$ are complementary to each other. When only $\mathcal{L}_F$ is added to GAIL, we observed an increase on HalfCheetah and Walker2d, not on Ant, and the gap in increase is small. Tian et al. (2020) demonstrated that it is not always good to learn maximal information using contrastive learning. Rather, it is important to minimize nuisance information as much as possible using a strong transformation to maximize task-specific information. However, a strong transformation without damaging semantic information is impossible in continuous control data. Thus, $\mathcal{L}_F$ mainly plays the role of maximally learning temporally predictive information with a weak, reasonable transformation, and $(\mathcal{L}_{SC}, \mathcal{L}_{AC})$ helps to suppress the nuisance factors using the corruption method so as to maximize task-relevant features in the proposed method.

Table 6: Ablation studies using 50 optimal and 50 non-optimal state-action pairs on Ant-v2 to test the role of a loss function of $\mathcal{L}_F$.

	$\mathcal{L}_F$
MSE	2793.48±98.59
BYOL	752.19±51.08
Barlow twins	3602.71±1164.86
SimCLR	**4384.71±49.21**

Table 7: Ablation studies using 50 optimal and 50 non-optimal state-action pairs on Ant-v2 to test the role of a loss function of $\mathcal{L}_{SC}$ and $\mathcal{L}_{AC}$. BT = Barlow twins.

		$\mathcal{L}_{SC}$					
		MSE	BYOL	SimSiam	BT	VICReg	Avg.
$\mathcal{L}_{AC}$	MSE	3658.97	-627.01	1516.06	2891.03	701.55	1628.12
	BYOL	2668.32	1240.80	1315.78	265.13	902.04	1278.41
	SimSiam	2717.81	4017.71	3854.55	3656.74	2943.91	3438.14
	BT	**4384.71**	2871.37	2590.40	3474.28	4269.20	**3517.99**
	VICReg	1927.16	2363.05	4027.86	3768.11	3458.10	3108.86
	Avg.	**3071.40**	1973.18	2660.93	2811.06	2454.96	

Sensitivity to corruption rate Recently, studies on the importance of the degree of transformation have been proposed in SSL. According to Tian et al. (2020), using previously suggested transformation techniques without searching can lead to an increase in learning nuisance information. Jing et al. (2021) showed that a strong transformation can cause dimensional collapse. They showed that a greedy search for finding appropriate corruption rates is important in applying SSL to RL benchmarks. Table 5 shows that the action is very vulnerable to a high corruption rate. For the state, the performance is maintained to some extent below 0.4. Additionally, we confirmed that fixing the corruption rate is better than providing a range.

Loss function of $\mathcal{L}_F$ For the discriminator D_ω, temporally predictive features are important information to distinguish the imitator (agent) from the expert. However, learning maximal information can lead to learning nuisance information as well. Transformation techniques should be used to suppress this. For $\mathcal{L}_F$, we tested the concatenation of Gaussian noise and the corrupted state as transformation techniques to generate the distorted version. The concatenation of Gaussian noise yields better results. Qualitatively, there was a 7.5% relative improvement when using the concatenation of Gaussian noise. We surmise that this is because corruption can unavoidably change important input features with respect to temporality. Also, we confirmed that appending noise dimensions rather than no noise concatenation increases the performance (Appendix E). For the loss function of $\mathcal{L}_F$, in addition to the InfoNCE loss of SimCLR, we tested MSE, and MSE with the stop-gradient of BYOL, and the Barlow twins loss. Table 6 presents that SimCLR's performance is superior to other methods by a significant margin.

Loss function of $\mathcal{L}_{SC}$ and $\mathcal{L}_{AC}$ As shown in Table 7, we tested various SSL loss functions for both $\mathcal{L}_{SC}$ and $\mathcal{L}_{AC}$. Notably, for $\mathcal{L}_{SC}$, the MSE loss that is exposed to the collapsing problem shows the highest performance on average. This is because $\mathcal{L}_F$ cannot be minimized if the state representation is only the same constant. Rather, the MSE loss that can flow the gradients to both the input pair shows a higher performance than the loss functions using a stop-gradient or different discrepancy measures. For $\mathcal{L}_{AC}$, the Barlow twins and SimSiam losses showed the first- and second-best performance on average, respectively. Because the role of the state is greater than that of the action when predicting the next state, the action representation z^a is not completely free from the collapsing problem. Therefore, unlike the trend of $\mathcal{L}_{SC}$, clearly, the loss functions that have a certain technique to prevent it showed stable performance.

7. Conclusion

Sample efficiency of expert demonstrations is desirable in imitation learning because obtaining a large number of expert demonstrations is often costly. Motivated by successes in self-supervised learning, we proposed a sample-efficient imitation learning method that promotes learning feature representations that are temporally predictive and robust against diverse distortions for continuous control. We evaluated our proposed method in various control tasks with limited expert demonstration settings and showed superior performance compared to existing methods. We analyzed the efficiency of the proposed method through both theoretical motivation and extensive experiments on continuous and discrete-control benchmarks.

Despite the excellent performance with limited settings, the proposed method has some limitations. There is an increase in model complexity and additional computational cost during training since three additional networks and self-supervised losses are added.

8. Acknowledgement

References

Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in Neural Information Processing Systems*, 34, 2021.

Saurabh Arora and Prashant Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, page 103500, 2021.

Dara Bahri, Heinrich Jiang, Yi Tay, and Donald Metzler. Scarf: Self-supervised contrastive learning using random feature corruption. *arXiv preprint arXiv:2106.15147*, 2021.

Paul Barde, Julien Roy, Wonseok Jeon, Joelle Pineau, Chris Pal, and Derek Nowrouzezahrai. Adversarial soft advantage fitting: Imitation learning without policy optimization. In *NeurIPS*, 2020.

Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

Andreea Bobu, Yi Liu, Rohin Shah, Daniel S Brown, and Anca D Dragan. Sirl: Similarity-based implicit representation learning. In *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, pages 565–574, 2023.

Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Daniel Brown, Russell Coleman, Ravi Srinivasan, and Scott Niekum. Safe imitation learning via fast bayesian reward inference from preferences. In *International Conference on Machine Learning*, pages 1165–1177. PMLR, 2020.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

Jiaao Chen, Zichao Yang, and Diyi Yang. Mixtext: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157, 2020a.

Minshuo Chen, Yizhou Wang, Tianyi Liu, Zhuoran Yang, Xingguo Li, Zhaoran Wang, and Tuo Zhao. On computation and generalization of generative adversarial imitation learning. In *International Conference on Learning Representations*, 2019.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020b.

Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

Ali Dabouei, Sobhan Soleymani, Fariborz Taherkhani, and Nasser M Nasrabadi. Supermix: Supervising the mixing data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13794–13803, 2021.

Tri Dao, Albert Gu, Alexander Ratner, Virginia Smith, Chris De Sa, and Christopher Ré. A kernel theory of modern data augmentation. In *International Conference on Machine Learning*, pages 1528–1537. PMLR, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

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

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Pires, Zhaohan Guo, Mohammad Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *Neural Information Processing Systems*, 2020.

Hongyu Guo, Yongyi Mao, and Richong Zhang. Augmenting data with mixup for sentence classification: An empirical study. *arXiv preprint arXiv:1905.08941*, 2019.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2019.

Nicklas Hansen, Rishabh Jangir, Yu Sun, Guillem Alenyà, Pieter Abbeel, Alexei A Efros, Lerrel Pinto, and Xiaolong Wang. Self-supervised policy adaptation during deployment. In *International Conference on Learning Representations*, 2020.

Nicklas Hansen, Hao Su, and Xiaolong Wang. Stabilizing deep q-learning with convnets and vision transformers under data augmentation. *Advances in Neural Information Processing Systems*, 34, 2021.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

Zhuoxun He, Lingxi Xie, Xin Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Data augmentation revisited: Rethinking the distribution gap between clean and augmented data. *arXiv preprint arXiv:1909.09148*, 2019.

Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.

Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. In *International Conference on Learning Representations*, 2019.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29:4565–4573, 2016.

Zehong Hu, Yitao Liang, Jie Zhang, Zhao Li, and Yang Liu. Inference aided reinforcement learning for incentive mechanism design in crowdsourcing. *Advances in Neural Information Processing Systems*, 31:5507–5517, 2018.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348*, 2021.

Kacper Piotr Kielak. Do recent advancements in model-based deep reinforcement learning really improve data efficiency?, 2020.

JangHyun Kim, Wonho Choo, Hosan Jeong, and Hyun Oh Song. Co-mixup: Saliency guided joint mixup with supermodular diversity. In *International Conference on Learning Representations*, 2020.

Misha Laskin, Kimin Lee, Adam Stooke, Lerrel Pinto, Pieter Abbeel, and Aravind Srinivas. Reinforcement learning with augmented data. *Advances in Neural Information Processing Systems*, 33, 2020.

Alex Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *Advances in Neural Information Processing Systems*, 33, 2020.

Kibok Lee, Yian Zhu, Kihyuk Sohn, Chun-Liang Li, Jinwoo Shin, and Honglak Lee. i-mix: A domain-agnostic strategy for contrastive representation learning. 2021.

Xiang Li, Jinghuan Shang, Srijan Das, and Michael S Ryoo. Does self-supervised learning really improve reinforcement learning from pixels? In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

Puneet Mangla, Nupur Kumari, Abhishek Sinha, Mayank Singh, Balaji Krishnamurthy, and Vineeth N Balasubramanian. Charting the right manifold: Manifold mixup for few-shot learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2218–2227, 2020.

Bogdan Mazoure, Remi Tachet des Combes, Thang Long DOAN, Philip Bachman, and R Devon Hjelm. Deep reinforcement and infomax learning. *Advances in Neural Information Processing Systems*, 33, 2020.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Shakir Mohamed and Danilo Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. *Advances in Neural Information Processing Systems*, 28:2125–2133, 2015.

Krikamol Muandet, Kenji Fukumizu, Francesco Dinuzzo, and Bernhard Schölkopf. Learning from distributions via support measure machines. In *Proceedings of the 25th International Conference on Neural Information Processing Systems-Volume 1*, pages 10–18, 2012.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.

Xue Bin Peng, Angjoo Kanazawa, Sam Toyer, Pieter Abbeel, and Sergey Levine. Variational discriminator bottleneck: Improving imitation learning, inverse RL, and GANs by constraining information flow. In *International Conference on Learning Representations*, 2019.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA ARTIFICIAL INTELLIGENCE AND PSYCHOLOGY . . . , 1989.

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180. PMLR, 2019.

Ahmed H Qureshi, Byron Boots, and Michael C Yip. Adversarial imitation via variational inverse reinforcement learning. In *International Conference on Learning Representations*, 2018.

Nathan D Ratliff, J Andrew Bagnell, and Martin A Zinkevich. Maximum margin planning. In *Proceedings of the 23rd international conference on Machine learning*, pages 729–736, 2006.

Siddharth Reddy, Anca D Dragan, and Sergey Levine. Sqil: Imitation learning via reinforcement learning with sparse rewards. In *International Conference on Learning Representations*, 2019.

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

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. PMLR, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*, 2020.

Riley Simmons-Edler, Ben Eisner, Daniel Yang, Anthony Bisulco, Eric Mitchell, Sebastian Seung, and Daniel Lee. Reward prediction error as an exploration objective in deep rl. *arXiv preprint arXiv:1906.08189*, 2019.

Avi Singh, Larry Yang, Kristian Hartikainen, Chelsea Finn, and Sergey Levine. End-to-end robotic reinforcement learning without reward engineering. *arXiv preprint arXiv:1904.07854*, 2019.

Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020.

Richard S Sutton and Andrew G Barto. Introduction to reinforcement learning, 1998.

Voot Tangkaratt, Nontawat Charoenphakdee, and Masashi Sugiyama. Robust imitation learning from noisy demonstrations. In *International Conference on Artificial Intelligence and Statistics*, pages 298–306. PMLR, 2021.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *arXiv preprint arXiv:2005.10243*, 2020.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

Faraz Torabi, Sean Geiger, Garrett Warnell, and Peter Stone. Sample-efficient adversarial imitation learning from observation. *arXiv preprint arXiv:1906.07374*, 2019.

Vladimir N Vapnik. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999, 1999.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR, 2019a.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR, 2019b.

Kaixin Wang, Bingyi Kang, Jie Shao, and Jiashi Feng. Improving generalization in reinforcement learning with mixture regularization. In *NeurIPS*, 2020.

Tianyu Wang, Nikhil Karnwal, and Nikolay Atanasov. Latent policies for adversarial imitation learning. *arXiv preprint arXiv:2206.11299*, 2022.

Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.

Keyu Wu, Min Wu, Zhenghua Chen, Yuecong Xu, and Xiaoli Li. Generalizing reinforcement learning through fusing self-supervised learning into intrinsic motivation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8683–8690, 2022.

Yueh-Hua Wu, Nontawat Charoenphakdee, Han Bao, Voot Tangkaratt, and Masashi Sugiyama. Imitation learning from imperfect demonstration. In *International Conference on Machine Learning*, pages 6818–6827. PMLR, 2019.

Denis Yarats, Ilya Kostrikov, and Rob Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations*, 2020.

Jinsung Yoon, Yao Zhang, James Jordon, and Mihaela van der Schaar. Vime: Extending the success of self-and semi-supervised learning to tabular domain. *Advances in Neural Information Processing Systems*, 33, 2020.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019.

Andrii Zadaianchuk, Georg Martius, and Fanny Yang. Self-supervised reinforcement learning with independently controllable subgoals. In *Conference on Robot Learning*, pages 384–394. PMLR, 2022.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stephane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12310–12320. PMLR, 18–24 Jul 2021.

Songyuan Zhang, Zhangjie Cao, Dorsa Sadigh, and Yanan Sui. Confidence-aware imitation learning from demonstrations with varying optimality. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

Xin Zhang, Yanhua Li, Ziming Zhang, and Zhi-Li Zhang. f-gail: Learning f-divergence for generative adversarial imitation learning. *Advances in neural information processing systems*, 2020.

Brian D Ziebart, Andrew Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd national conference on Artificial intelligence-Volume 3*, pages 1433–1438, 2008a.

Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008b.

Appendix A. Proof for Section 3

A.1 Proof of Corollary 1

Using the first-order Taylor approximation, we can expand $\hat{R}(f_{h_{\mathrm{MSE}}}|\mathcal{P})$.

Theorem 1. *Based on Dao et al. (2019), picking up* $\mathbf{z}_0 = \underset{\mathbf{z}_i \sim h_{MSE}(X_i)}{\mathbb{E}}[\mathbf{z}_i]$, *the first-order approximation of* $\hat{R}(f_{h_{MSE}}|\mathcal{P})$ *is* $\frac{1}{M}\sum_{i=1}^{M} \ell\left(\mathbf{W}^{\top} \underset{\mathbf{z}_i \sim h_{MSE}(X_i)}{\mathbb{E}}[\mathbf{z}_i], y_i\right)$. *Then, the first-order effect is that it averages the distorted features that are not necessarily present in the original dataset* $\mathcal{P}$.

Proof.   For proof, we define a kernel classifier K to map a feature space. The kernel is always positive definite, and its space is expressed through a dot product. The kernel defines various functions, and its space of such functions is called the reproducing kernel Hilbert space (RKHS).

When we define that $T(x)$ is the probability density that x inherently contains a distortion that is not necessarily present in the original dataset and a kernel K' represents a new feature space via Eq. 4, we can have the following proof.

$$
\begin{aligned}
K'(x, \tilde{x}) &= \left\langle \mathbb{E}_{x \sim T(x)}[h_{\mathrm{MSE}}(x)], \mathbb{E}_{\tilde{x} \sim T(\tilde{x})}[h_{\mathrm{MSE}}(\tilde{x})] \right\rangle \\
&= \int_{x \in \mathbb{R}^n} \int_{\tilde{x} \in \mathbb{R}^n} \left\langle h_{\mathrm{MSE}}(x), h_{\mathrm{MSE}}(\tilde{x}) \right\rangle T(x)T(\tilde{x}) d\tilde{x}dx \\
&= \int_{x \in \mathbb{R}^n} \int_{\tilde{x} \in \mathbb{R}^n} K(x, \tilde{x})T(x)T(\tilde{x}) d\tilde{x}dx \\
&= (TKT^{\top})(x, \tilde{x}),
\end{aligned}
$$

where x and $\tilde{x}$ are two inputs. It shows that the proposed approach can be interpreted as a linear classifier $\mathbf{W}$ on the new kernel K'. This proof is similar to how a support measure machine (SMM) trains the model with a mean function in the RKHS (Muandet et al., 2012). SMM is proved to be invariant on the distorted features that are not necessarily present in the original dataset $\mathcal{P}$. In the newly learned feature space through Eq. 4, the distorted features, which are inherently contained in the original dataset, become non-informative. As a result, the sample complexity required to learn the classifier $f_{h_{\mathrm{MSE}}}$ is reduced. More formally, a support vector machine (SVM) (Hearst et al., 1998) is a special case of SMM. SMM is effective at seeking a halfspace that separates a training set by a large margin. In other words, our proposed approach similar to SMM is also effective in inducing the low VC dimension.

Corollary 1. h_{MSE} *is effective in reducing the VC dimension, however, the major problem is that the upper bound of* $R(f|\mathcal{P})$ *we seek to is based on* $\hat{R}(f|\mathcal{P})$ *instead of* $\hat{R}(f_{h_{MSE}}|\mathcal{P})$. *As a result,* $f_{h_{MSE}}$ *cannot capture all the properties preserved in* $\hat{R}(f|\mathcal{P})$. *Additionally, due to the distribution gap between the feature spaces* h *and* h_{MSE}, *the optimal function* $f_{h_{MSE}}^{*}$ *of* $\hat{R}(f_{h_{MSE}}|\mathcal{P})$ *is not guaranteed to be a minimum of* $\hat{R}(f|\mathcal{P})$.

He et al. (2019) showed that the optimal function f^* of $\hat{R}(f|\mathcal{P})$ leverages both class-relevant and -irrelevant features ($w_{d,i} \approx w_{d,j}$ for $1 \leq d \leq D$ and $1 \leq i, j \leq C$) (Definition 3). However, when finding $f^*_{h_{\text{MSE}}}$, distorted, commonly class-irrelevant, features are restrained to be encoded through Eq. 4. Thus, the optimal function $f^*_{h_{\text{MSE}}}$ of $\hat{R}(f_{h_{\text{MSE}}}|\mathcal{P})$ is not guaranteed to be a minimum of $\hat{R}(f|\mathcal{P})$ due to a distribution shift.

A.2 Proof of Claim 1

Suppose that the MSE (Eq. 4) and contrastive learning share its feature space $\mathcal{H}$.

Definition 1. *(Wang and Isola, 2020) Consider a perfectly trained h^*_{MSE} with unsupervised contrastive learning. A positive pair has similar features on $\mathcal{H}_{MSE}$ (Alignment), and the feature space $\mathcal{H}_{MSE}$ becomes maximally informative (Uniformity).*

Definition 2. *(Poole et al., 2019) Contrastive loss is defined using a critic function that approximates density ratios ($\frac{p(x|y)}{p(x)} = \frac{p(y|x)}{p(y)}$) of two random variables X and Y. It is proved that contrastive loss is a lower bound on mutual information (MI), $I(X;Y)$. By minimizing the loss, the lower bound on MI is maximized.*

Unsupervised contrastive learning can be interpreted as an InfoMax principle, as shown in Definitions 1 and 2. In our terms, minimizing contrastive loss is maximizing the lower bound on $I(h_{\text{MSE}}(x); h_{\text{MSE}}(x'))$ for a positive pair $(x, x') \sim \mathcal{P}_{\text{pos}}$. MI between $h_{\text{MSE}}(x)$ and $h_{\text{MSE}}(x')$ can be re-written as follows:

$$I(h_{\text{MSE}}(x); h_{\text{MSE}}(x')) = H(h_{\text{MSE}}(x)) - H(h_{\text{MSE}}(x)|h_{\text{MSE}}(x')).$$

The first term of the right-hand side (RHS) shows that uniformity in Definition 1 is mathematically favored by entropy $H(h_{\text{MSE}}(x))$. The second term of RHS shows that alignment in Definition 1 is mathematically favored by conditional entropy $H(h_{\text{MSE}}(x)|h_{\text{MSE}}(x'))$. As a result, to maximally increase the closeness between all positive pairs in the feature space, all informative features should be semantically embedded and utilized. That is, mapping more informative features induces that more similar inputs have more similar features in $\mathcal{H}$. Through that, informative yet class-irrelevant features are also preserved for contrastive learning.

Theorem 2. *Based on Dao et al. (2019), with $\psi = \mathbb{E}_{z \sim h_{MSE}(X)}[z]$ and $Var = \mathbb{E}\left[(X - \mathbb{E}[X])^2\right]$, the second-order approximation of $\hat{R}(f_{h_{MSE}}|\mathcal{P})$ is the first-order approximation plus variance regularization of weight vectors w_d for $1 \leq d \leq D$.*

Proof. For cross entropy loss with softmax, ℓ'' is independent of y and positive semi-definite. We can obtain a more exact expression for $\hat{R}(f_{h_{\text{MSE}}}|\mathcal{P})$ by the second-order approximation of the Taylor expansion.

$$\hat{R}(f_{h_{\mathrm{MSE}}}|\mathcal{P}) = \frac{1}{M}\sum_{i=1}^{M}\ell\left(\mathbf{W}^{\top}\psi_i, y_i\right) + \frac{1}{2M}\sum_{i=1}^{M}\mathop{\mathbb{E}}_{z_i\sim h_{\mathrm{MSE}}(X_i)}\left[\left(\mathbf{W}^{\top}(z_i-\psi_i)\right)^2\ell''(\mathbf{W}^{\top}z_i, y_i)\right]$$

$$= \frac{1}{M}\sum_{i=1}^{M}\ell\left(\mathbf{W}^{\top}\psi_i, y_i\right) +$$

$$\mathbf{W}^{\top}\left(\frac{1}{2M}\sum_{i=1}^{M}\mathop{\mathbb{E}}_{z_i\sim h_{\mathrm{MSE}}(X_i)}\left[\Delta_i\Delta_i^{\top}\ell''(\mathbf{W}^{\top}z_i, y_i)\right]\right)\mathbf{W} \quad (\because \Delta_i = z_i - \psi_i)$$

$$= \frac{1}{M}\sum_{i=1}^{M}\ell\left(\mathbf{W}^{\top}\psi_i, y_i\right) +$$

$$\frac{1}{2M}\sum_{i=1}^{M}\mathbf{W}^{\top}\mathop{\mathbb{E}}_{z_i\sim h_{\mathrm{MSE}}(X_i)}\left[\Delta_i\Delta_i^{\top}\right]\ell''(\mathbf{W}^{\top}\psi_i)\mathbf{W} \quad (\because \ell'' \text{ is independent of } y)$$

The second-order approximation of $\hat{R}(f_{h_{\mathrm{MSE}}}|\mathcal{P})$ can be interpreted as the first-order approximation plus variance regularization of weight vectors w_d for $1 \leq d \leq D$ because $\Delta\Delta^{\top}$ is equal to the variance of z. It represents that the weights of features having large variances are forced to 0 restricted by regularization. We do not utilize data augmentation in the classification loss because it further induces the generalization gap. When the feature space is learned through only Eq. 4 and the classification loss, most class-irrelevant features cannot be encoded in the feature manifold. We tackle this problem by mapping class-irrelevant yet informative features learned from contrastive learning to the feature manifold. For contrastive learning, the positive pair, which is created by two differently augmented same images, should have very similar feature representation. It means informative features mapped through this loss are robust to the variance via augmentation.

Claim 1. *Class-irrelevant yet informative features are preserved via unsupervised contrastive learning. It reduces the inconsistency between $\hat{R}(f_{h_{MSE}}|\mathcal{P})$ and $\hat{R}(f|\mathcal{P})$.*

Class-irrelevant yet informative features are preserved via unsupervised contrastive learning. In other words, the feature manifold is enriched by unsupervised contrastive learning, that is, both class-relevant and class-irrelevant yet informative features can be leveraged when finding the optimal function $f_{h_{\mathrm{MSE}}}^{*}$ of $\hat{R}(f_{h_{\mathrm{MSE}}}|\mathcal{P})$. As a result, the inconsistency between $\hat{R}(f_{h_{\mathrm{MSE}}}|\mathcal{P})$ and $\hat{R}(f|\mathcal{P})$ can be reduced.

Appendix B. Additional Implementation and Experimental Details

We use 5 continuous control benchmarks on Mujoco (Todorov et al., 2012) (Ant-v2, Hopper-v2, HalfCheetah-v2, Swimmer-v2, and Walker2d-v2), and 2 discrete control benchmarks on Atari RAM (BeamRider-ram-v0, and SpaceInvaders-ram-v0) of OpenAI Gym (Brockman et al., 2016).

Overall, we reported the mean and standard error of the performance over 3 trials. For experimental settings, we used GTX 1080 Ti for GPUs, Intel i7-6850K for CPUs, and Ubuntu 18.04 for OS. The usage of GPU memory is approximately 3000MB for training,

and training time for tested benchmarks was approximately 20 hours. Our code is based on Pytorch (Paszke et al., 2019) and python libraries.

We make use of the same neural net architecture and hyperparameters for all benchmarks. For the policy network, value network, discriminator, and state encoder, we use 3 hidden layers with size 100 and Tanh as activation functions. For the action encoder, we use 6 1D convolutional layers with sizes (64, 64, 64, 128, 256, and 256), 1 hidden layer with size 8 as the output, and LeakyReLU as activation functions. For the forward dynamics model, we use 1 hidden layer with size 114 and ReLU as activation functions.

For hyperparameters in all runs, the total epoch for Swimmer, and Hopper is 3,000, for BeamRider, SpaceInvaders, HalfCheetah, and Walker2d is 5,000, and for Ant is 8,000. Please refer to Table 8 for other hyperparameters. We set $\lambda_F = 1$, $\lambda_S = 100$, and $\lambda_A = 1$ for matching loss scale.

Table 8: Base hyperparameters used for all benchmarks.

Hyperparameters	Value
γ	0.995
Generalized advantage estimation	0.97
N	5,000
Learning rate (all networks except for value network)	1e-3
Learning rate (value network)	3e-4
Batch size (RERP)	256
Batch size (TRPO)	128
Batch size (GAIL)	5,000
Optimizer (all networks)	Adam
τ	0.1
λ_F	1
λ_S	100
λ_A	1

For the random corruption method (Yoon et al., 2020), we sampled an imputing value from $\mathcal{N}(0, 1)$ by considering the range of state values. The mean value used for imputation (Bahri et al., 2021) is the mean vector of each $\mathcal{D}_k$ where k is the training iteration. We imputed with a mean value of the corresponding dimension of the calculated mean vector.

B.1 Optimality of 100% (Expert Demonstrations)

To train experts, we used a reinforcement learning algorithm, proximal policy optimization (Schulman et al., 2017), uploaded in the official Github by the authors of CAIL (Zhang et al., 2021). We selected the converged policy as the expert policy. The performance of the utilized expert policy and other specifications related to the experiments on the main manuscript are given in Table 9.

B.2 Optimality of 25%, 50%, or 75% (A Mixture of Optimal and Non-optimal Demonstrations)

We also tested the effectiveness of the proposed method with imperfect demonstrations. We combined the proposed method with the existing method for imperfect demonstrations

Table 9: Specification and the number of used demonstrations of each continuous control benchmark in the scenario of perfect expert demonstrations.

Benchmarks	$\dim(S)$	$\dim(A)$	N_E	Expert's Performance
HalfCheetah-v2	$\mathbb{R}^{17}$	$\mathbb{R}^6$	100	5455.49±74.26
Walker-v2	$\mathbb{R}^{17}$	$\mathbb{R}^6$	100	3685.27±57.99
Ant-v2	$\mathbb{R}^{111}$	$\mathbb{R}^8$	100	4787.23±115.72

and checked improvement in performance. We compared our combined method against the following baselines: BC (Pomerleau, 1989), GAIL (Ho and Ermon, 2016), IC-GAIL (Wu et al., 2019), 2IWIL (Wu et al., 2019), RIL-CO (Tangkaratt et al., 2021), and CAIL (Zhang et al., 2021).

IL methods, especially variants of BC, require a large volume of expert demonstration data for training (Ross et al., 2011). These methods struggle with a generalization problem when using a small number of demonstrations. Empirically, we observed that BC almost fails to converge on Ant, Walker2d, and Hopper with a small number of demonstrations. For RIL-CO, they measure and optimize a classification risk with the symmetric loss. Basically, they only assumed a scenario that the majority of demonstrations are obtained using an optimal policy. IC-GAIL and 2IWIL were proposed in the same paper. Both methods are confidence-based. In the case of 2IWIL, the confidence of each state-action pair is estimated using the classification risk before training, and in the case of IC-GAIL, they implicitly utilize the confidence score in a way of matching the occupancy measure of the imitator with the expert. CAIL is the state-of-the-art work in IL algorithms for imperfect demonstrations. They jointly learn the confidence score and policy using an outer loss. Because they update two factors jointly, the training is unstable. Experimentally, there was no benchmark that CAIL, which is most recently suggested, is superior to 2IWIL or RIL-CO with a small number of imperfect demonstrations. We surmise that this is because CAIL is the method using a full trajectory rather than each state-action pair when estimating the confidence score of each pair. Consequently, we decided to combine our method with 2IWIL instead of CAIL.

Table 10: Specification and the number of used demonstrations of each continuous control benchmark in the scenario of imperfect expert demonstrations.

Benchmarks	$\dim(S)$	$\dim(A)$	N_E	Suboptimal 1	Suboptimal 2	Suboptimal 3	Suboptimal 4	Expert's Performance
HalfCheetah-v2	$\mathbb{R}^{17}$	$\mathbb{R}^6$	100	1051.91±50.17	2280.87±651.92	3533.07±79.47	4682.89±54.33	5455.49±74.26
Walker-v2	$\mathbb{R}^{17}$	$\mathbb{R}^6$	100	691.14±96.12	1617.02±721.00	2579.41±512.19	2819.63±609.49	3685.27±57.99
Ant-v2	$\mathbb{R}^{111}$	$\mathbb{R}^8$	100	789.13±170.45	2115.17±328.20	2947.49±191.72	3739.91±96.56	4787.23±115.72
Swimmer-v2	$\mathbb{R}^8$	$\mathbb{R}^2$	20	65.56±18.93	148.20±8.09	181.27±4.23	228.29±5.95	280.5±1.24
Hopper-v2	$\mathbb{R}^{11}$	$\mathbb{R}^3$	20	1262.34±296.32	1774.65±462.52	2185.33±996.92	2802.18±489.85	3531.03±23.51

For the experiments on the main manuscript, we collected a mixture of optimal and non-optimal demonstrations with different optimalities. For collecting the imperfect demonstrations, we used the official Github by the authors of CAIL. Following CAIL, We selected four intermediate policies as sub-optimal policies and the converged policy as the optimal policy. The performance of sub-optimal and optimal policies and other specifications for the benchmarks are given in Table 10. Empirically, as shown in Table 10, because the dimension

of the action space of Swimmer-v2 and Hopper-v2 is very small almost like discrete control, we did not apply $\mathcal{L}_{AC}$.

The pseudo-code of the combined methods is given in Algorithms 2 and 3. To run the combined methods, we need additional hyperparameters and their values are summarized in Table 11.

Table 11: Additional hyperparameters for the combined methods.

Hyperparameters	Value
α for Mixup	4.0
Threshold for GMM	0.5
The number of non-experts	4
Ratio of labeled demonstrations	0.4

Appendix C. Pseudo-code of Combined Method

C.1 Combined Case #1: Ours + 2IWIL

Because labeling all state-action pairs from ρ_O or ρ_N can be expensive, following previous IL works for imperfect demonstrations, we assumed that only a few demonstrations are labeled. Then, the imperfect demonstrations are split into two demonstrations: a set of labeled demonstrations $\mathcal{D}_L = \{(x_{l,i}, y_{l,i})\}_{i=1}^{N_L}$ and a set of unlabeled demonstrations $\mathcal{D}_U = \{x_{u,i}\}_{i=1}^{N_U}$, where N_L and N_U are the number of labeled demonstrations and unlabeled demonstrations, respectively.

As shown in Algorithm 2, before training, 2IWIL (Wu et al., 2019) estimates pseudo labels $\hat{y}_{u,i}$ of $\mathcal{D}_U$ to utilize the set of unlabeled demonstrations $\mathcal{D}_U$ with $\mathcal{D}_L$ using the classification risk proposed in their work as follows:

$$R_{\ell(g)} = \mathbb{E}_{x,\,y\,\sim\,\mathcal{D}_l}\left[y(\ell(g(x)) - \ell(-g(x))) + (1-\beta)\ell(-g(x))\right] + \mathbb{E}_{x\,\sim\,\mathcal{D}_u}\left[\beta\ell(-g(x))\right], \quad (16)$$

where $\beta = \frac{N_U}{N_L + N_U}$, $g(\cdot)$ is a neural network classifier, and ℓ is a strictly proper composite loss. Then, the predicted confidence score $\hat{y}_u$ represents the probability that a given state-action pair is optimal. The estimated confidence of each state-action pair is utilized as a sample weight in their discriminator loss. As a result, the combined discriminator loss with our state and action encoders is defined as follows:

$$\max_{\omega} \mathbb{E}_{x\sim\mathcal{D}_\pi}\left[\log D_\omega(z)\right] + \mathbb{E}_{(x,y)\,\sim\,(\tilde{\mathcal{D}}_O \cup \tilde{\mathcal{D}}_N)}\left[\frac{y}{\epsilon}\log(1 - D_\omega(z))\right], \quad (17)$$

where $\epsilon = \frac{1}{N_L}\sum_{i=1}^{N_L} y_i$, $z = z^s \oplus z^a$. z^s is a state representation embedded by $SE(s)$, and z^a is an action representation embedded by $AE(a)$.

C.2 Combined Case #2: Ours + 2IWIL + Manifold Mixup

For utilizing Manifold mixup (MM), we modeled the per-sample confidence score distribution of $(y_l, \hat{y}_u)$ with a two-component Gaussian mixture model to divide the imperfect demonstrations $\mathcal{D}_I$ into optimal demonstrations and non-optimal demonstrations, $\tilde{\mathcal{D}}_O$ and $\tilde{\mathcal{D}}_N$. The

---

Algorithm 2 Pseudo-code of Ours + 2IWIL (Wu et al., 2019)

1: **input**: Imperfect expert demonstrations $\mathcal{D}_I = \{\mathcal{D}_L \cup \mathcal{D}_U\}$, Labeled demonstrations $\mathcal{D}_L \triangleq \{(x_{l,i}, y_{l,i})\}_{i=1}^{N_L}$, Unlabeled demonstrations $\mathcal{D}_U \triangleq \{x_{u,i}\}_{i=1}^{N_U}$, # of batches B, Training epochs T.

2: Train a probabilistic classifier by minimizing Eq. 16

3: Predict confidence scores $\{\hat{y}_{u,i}\}_{i=1}^{N_u}$ for $\{x_{u,i}\}_{i=1}^{N_u}$

4: **for** $k \leftarrow 1$ to T **do**

5:     Obtain trajectories $\mathcal{D}_k = \{x_{k,i}\}_{i=1}^{N}$ using π_θ

6:     $\pi_\theta \leftarrow \text{TRPO}(\pi_\theta, V, D_\omega, \mathcal{D}_k)$

7:     $SE, AE \leftarrow \text{REPR}(SE, AE, F, \mathcal{D}_k)$

8:     $D_\omega \leftarrow \text{GAIL}(D_\omega, SE, AE, \mathcal{D}_k, \mathcal{D}_I)$

9: **end for**

10: **function** REPR$(SE, AE, F, \mathcal{D}_k)$

11:     **for** $b \leftarrow 1$ to B **do**

12:         Generate X_b' by Eq. 8

13:         Obtain Z_b from $\mathcal{D}_{k,b}$ using (SE, AE)

14:         Obtain Z_b' from X_b' using (SE, AE)

15:         Update SE, AE, and F by Eq. 14

16:     **end for**

17:     **return** SE, AE

18: **end function**

19: **function** GAIL$(D_\omega, SE, AE, \mathcal{D}_k, \mathcal{D}_I)$

20:     **for** $b \leftarrow 1$ to B **do**

21:         Obtain Z_b from $\mathcal{D}_{k,b}$ using (SE, AE)

22:         Update SE, AE, and D_ω by Eq. 17

23:     **end for**

24:     **return** D_ω

25: **end function**

---

feature representation $z_o \sim \tilde{\mathcal{D}}_O$ is interpolated with the feature representation $z_n \sim \tilde{\mathcal{D}}_N$. More formally, for a batch of features (z_o, z_n) and corresponding confidence scores (y_o, y_n), the mixed $(\bar{z}, \bar{y})$ can be computed by:

$$
\begin{aligned}
\lambda &\sim \text{Beta}(\alpha, \alpha), \quad \lambda' = \max(\lambda, 1 - \lambda), \\
\bar{z} &= \lambda' \cdot z_o + (1 - \lambda') \cdot z_n, \\
\bar{y} &= \lambda' \cdot y_o + (1 - \lambda') \cdot y_n,
\end{aligned}
\tag{18}
$$

where $z = z^s \oplus z^a$. z^s is a state representation embedded by $SE(s)$, z^a is an action representation embedded by $AE(a)$. Eq. 18 can ensure that $\bar{z}$ are closer to optimal demonstrations than non-optimal demonstrations.

Algorithm 3 Pseudo-code of Ours + 2IWIL + Manifold Mixup (Verma et al., 2019a)

1: **input**: Imperfect expert demonstrations $\mathcal{D}_I = \{\mathcal{D}_L \cup \mathcal{D}_U\}$, Labeled demonstrations $\mathcal{D}_L \triangleq \{(x_{l,i}, y_{l,i})\}_{i=1}^{N_L}$, Unlabeled demonstrations $\mathcal{D}_U \triangleq \{x_{u,i}\}_{i=1}^{N_U}$, # of batches B, Training epochs T.
2: Train a probabilistic classifier by minimizing Eq. 16
3: Predict confidence scores $\{\hat{y}_{u,i}\}_{i=1}^{N_U}$ for $\{x_{u,i}\}_{i=1}^{N_U}$
4: $\tilde{\mathcal{D}}_O, \tilde{\mathcal{D}}_N \leftarrow GMM(\mathcal{D}_L, (\mathcal{D}_U, \{\hat{y}_{u,i}\}_{i=1}^{N_U}))$
5: **for** $k \leftarrow 1$ to T **do**
6:     Obtain trajectories $\mathcal{D}_k = \{x_{k,i}\}_{i=1}^N$ using π_θ
7:     $\pi_\theta \leftarrow \text{TRPO}(\pi_\theta, V, D_\omega, \mathcal{D}_k)$
8:     $SE, AE \leftarrow \text{REPR}(SE, AE, F, \mathcal{D}_k)$
9:     $D_\omega \leftarrow \text{GAIL}(D_\omega, SE, AE, \mathcal{D}_k, \tilde{\mathcal{D}}_O, \tilde{\mathcal{D}}_N)$
10: **end for**
11: **function** REPR$(SE, AE, F, \mathcal{D}_k)$
12:     **for** $b \leftarrow 1$ to B **do**
13:         Generate X'_b by Eq. 8
14:         Obtain Z_b from $\mathcal{D}_{k,b}$ using (SE, AE)
15:         Obtain Z'_b from X'_b using (SE, AE)
16:         Update SE, AE, and F by Eq. 14
17:     **end for**
18:     **return** SE, AE
19: **end function**
20: **function** GAIL$(D_\omega, SE, AE, \mathcal{D}_k, \tilde{\mathcal{D}}_O, \tilde{\mathcal{D}}_N)$
21:     **for** $b \leftarrow 1$ to B **do**
22:         Obtain Z_b from $\mathcal{D}_{k,b}$ using (SE, AE)
23:         Obtain (Z_o, Z_n) from $(\tilde{\mathcal{D}}_O, \tilde{\mathcal{D}}_N)$ using (SE, AE)
24:         Compute $(\bar{Z}, \bar{Y})$ by Eq. 18
25:         Update SE, AE, and D_ω by Eq. 19
26:     **end for**
27:     **return** D_ω
28: **end function**

As shown in Algorithm 3, by additionally including synthetic data through MM, the discriminator loss is expressed as follows:

$$\max_\omega \mathbb{E}_{x \sim \mathcal{D}_\pi} \left[\log D_\omega(z)\right] + \mathbb{E}_{(x,y) \sim (\tilde{\mathcal{D}}_O \cup \tilde{\mathcal{D}}_N)} \left[\frac{y}{\epsilon}\log(1 - D_\omega(z))\right] +$$
$$\mathbb{E}_{(\bar{z},\bar{y}) \sim (\tilde{\mathcal{D}}_O \cup \tilde{\mathcal{D}}_N)} \left[\log((1 - \bar{y}) \cdot D_\omega(\bar{z}) + \bar{y} \cdot (1 - D_\omega(\bar{z})))\right]. \tag{19}$$

Appendix D. Expert Data Size

We assessed our method with varying expert data sizes. As shown in the table, there is a relatively small or no decrease in the performance up to $N_E = 20$. When N_E is reduced

to 20 for Ant and Walker2d and 10 for HalfCheetah, the performance is comparable to the baseline AIL.

Table 12: Additional studies using 10, 20, or 50 expert state-action pairs on Ant-v2, HalfCheetah-v2, and Walker2d-v2 of MuJoCo.

	GAIL	Ours			
	$N_E = 100$	$N_E = 10$	$N_E = 20$	$N_E = 50$	$N_E = 100$
Ant	4198.2±72.6	2855.9±1139.7	4286.4±125.6	4432.4±41.1	4554.8±162.6
HalfCheetah	2034.6±2384.6	2787.6±3454.4	5381.6±81.3	5410.6±44.2	5416.0±203.8
Walker2d	3513.4±172.9	3023.1±439.6	3412.6±195.6	3520.9±108.5	3527.6±131.4

Appendix E. Role of Gaussian Noise

Appending noise dimensions increases the performance on all three tasks as shown in Table 13. -71.09, -81.29, and –87.90 represent the decrease without appending the noise dimensions.

Table 13: Ablation studies on appending Gaussian noise using 100 expert state-action pairs on Ant-v2, HalfCheetah-v2, and Walker2d-v2 of MuJoCo.

Ours	Ant	HalfCheetah	Walker2d
w/o noise	4483.7±159.6 (-71.09)	5334.7±43.0 (-81.29)	3439.7±122.2 (-87.90)
w noise	4554.8±162.6	5416.0±203.8	3527.6±131.4

Appendix F. Discrete control Benchmarks

To test the scalability of the proposed method, we evaluated the performance of the proposed method on 2 discrete control benchmarks: BeamRider-ram-v0, and SpaceInvaders-ram-v0 of OpenAI Gym.

Table 14: Final performance using 20 expert state-action pairs on BeamRider-ram-v0, and SpaceInvaders-ram-v0 of OpenAI Gym. Best results are in **bold**.

BeamRider		SpaceInvaders	
GAIL	Ours	GAIL	Ours
399.43±55.63	**433.04±76.15**	166.39±107.88	**289.87±5.37**

For comparison, we chose GAIL, which showed the second-best performance in the experiments of continuous control benchmarks. To apply the proposed method on discrete control benchmarks, we ignored the action encoder AE and its corresponding loss $\mathcal{L}_{AC}$ of the proposed model. Nevertheless, as shown in Table 14, the proposed method still showed better performance compared to GAIL on discrete control benchmarks.