Sample-efficient Adversarial Imitation Learning

Dahuin Jung **Hyungyu Lee** **Sungroh Yoon**[*]

Electrical and Computer Engineering
Interdisciplinary Program in Artificial Intelligence
Seoul National University, Seoul 08826, Republic of Korea
{annajung0625,rucy74,sryoon}@snu.ac.kr

Abstract

Imitation learning, wherein 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. Particularly, in comparison with existing self-supervised learning methods for tabular data, we propose a different corruption method for state and action representations robust to diverse distortions. The proposed method shows a 39$\%$ relative improvement over the 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 the intuitions of a range of factors.

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 [39] 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 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) [36, 44]. However, error accumulation limits BC because it greedily imitates the demonstrated actions. The second approach is inverse reinforcement learning (IRL) [3], inferring a cost function based on given expert demonstrations. The IRL implements adversarial learning [16] 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.

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

[*] Correspondence to: Sungroh Yoon sryoon@snu.ac.kr.

Deep Reinforcement Learning Workshop at 36th Conference on Neural Information Processing Systems (NeurIPS 2022).

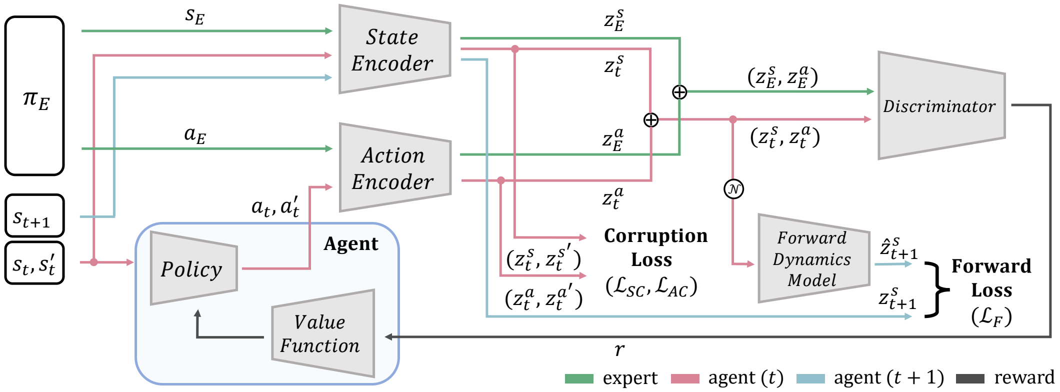

Figure 1: Overview of the proposed model. Details of each component and loss in the figure are described in Section 2.

address the first challenge [10, 15, 35, 38, 39]; however, little work has been done to improve the sample efficiency of expert demonstrations required [5, 55].

Self-supervised representation learning (SSL) has advanced sample efficiency in the image and language domains [11, 17, 31]. 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 [11, 21] and asymmetric twin-based [12, 17] 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 [41, 43] 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.

Moreover, learning representations that discard information regarding nuisance variables improves generalization and decreases required sample complexity. Previous transformation techniques for tabular data [4, 52] 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.

2 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:

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

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 $\hat{z}^s_{t+1}$ from the feature representations of the current state and action, z^s_t and z^a_t, and Gaussian noises $\mathcal{N}$. $\hat{z}^s_{t+1} = F(z^s_t \oplus z^a_t \oplus \mathcal{N}) \in \hat{Z}_{t+1}$, 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 in the Supplementary S2, the proposed method comprises three major parts. In Section 2.1, we explain how to train the cost function D_ω using expert demonstrations $\mathcal{D}_E$ (GAIL in Algorithm 1). In Section 2.2, we describe how to use SSL in a non-image environment (REPR in Algorithm 1). We implement trust region policy optimization [40] to train the agent policy π_θ by following the use in [22] (TRPO in Algorithm 1).

2.1 Generative Adversarial Imitation Learning

The proposed method is based on generative adversarial imitation learning (GAIL) [22]. 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 [16]. The minimax optimization of GAIL is expressed as follows:

$$\min_\theta \max_\omega \mathop{\mathbb{E}}_{x\sim\mathcal{D}_\pi} [\log D_\omega(x)] + \mathop{\mathbb{E}}_{x\sim\mathcal{D}_E} [\log(1 - D_\omega(x))] \,, \tag{2}$$

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 loss is expressed as follows:

$$\max_\omega \mathop{\mathbb{E}}_{x\sim\mathcal{D}_\pi} [\log D_\omega(z)] + \mathop{\mathbb{E}}_{x\sim\mathcal{D}_E} [\log(1 - D_\omega(z))] \,, \tag{3}$$

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 $AE(a)$.

2.2 State and Action Representations

2.2.1 Modeling forward Dynamics

Each domain requires different ways of generating self-supervision depending on the properties of the data. For example, BERT [14] 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 [34, 41]. 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:

$$\hat{z}^s_{t+1} = F(z^s_t \oplus z^a_t \oplus \mathcal{N}), \tag{4}$$

where $z^s_t = SE(s_t)$, $z^a_t = AE(a_t)$ at a time step t, and $\mathcal{N}(0, 1)$ is the Gaussian noise. The output $\hat{z}^s_{t+1}$ 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 [37, 49]. 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:

$$\mathcal{L}_F = -\mathbb{E}\left[\log \frac{e^{\mathrm{cs}(\hat{z}^s_{i,t+1}, z^s_{i,t+1})/\tau}}{\zeta}\right], \quad \zeta = \sum_{j=1}^{BS} \mathbb{1}_{j \neq i} e^{\mathrm{cs}(\hat{z}^s_{i,t+1}, z^s_{j,t})/\tau} + \sum_{j=1}^{BS} \mathbb{1}_{j \neq i} e^{\mathrm{cs}(\hat{z}^s_{i,t+1}, z^s_{j,t+1})/\tau}, \tag{5}$$

where $(z^s_t, z^s_{t+1}, \hat{z}^s_{t+1}) \sim (Z^s_t, Z^s_{t+1}, \hat{Z}^s_{t+1})$ and $\mathrm{cs}(u, v) = u^\top v / ||u||\, ||v||$ (j indexes the state or next state representation in the batch, and BS is the batch size). Following SimCLR [11], we use the other $2(BS - 1)$ representations in the batch as negative samples.

2.2.2 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^c_b and permute X^c_b by changing the order of each state-action pair in the batch at random, $perm(X^c_b)$. 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:

$$\begin{aligned} X^c_b[I] &= perm(X^c_b)[I], \\ X'_b &= X^c_b, \end{aligned} \tag{6}$$

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:

$$\mathcal{L}_{SC} = \mathbb{E}\left\| z^s - z^{s'} \right\|_2^2, \tag{7}$$

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. For the action representation, because the gradients from the $\mathcal{L}_F$ are not sufficient to hinder collapse, we use the Barlow twins loss [53] that is robust against the constant embedding problem. The Barlow twins loss draws the cross-correlation matrix close to the identity matrix. This is expressed as follows:

$$\mathcal{L}_{AC} = \sum_i (1 - \mathcal{C}_{ii})^2 - \sum_i \sum_{j \neq i} {\mathcal{C}_{ij}}^2, \qquad \mathcal{C}_{ij} = \frac{Z_i^a Z_j^{a'}}{\sqrt{(Z_i^a)^2}\sqrt{(Z_i^{a'})^2}} \tag{8}$$

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 5. Consequently, the total loss for SSL is computed as follows.

$$\mathcal{L}_{SS} = \lambda_F \mathcal{L}_F + \lambda_S \mathcal{L}_{SC} + \lambda_A \mathcal{L}_{AC}, \tag{9}$$

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

3 Related Work

3.1 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 [18, 27]. 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 [32, 34, 42]. For the image state, various image augmentation techniques and self-supervised objectives have been applied to reduce environmental interactions [19, 20, 25, 26, 30, 41, 43, 51]. To the best of our knowledge, this is the first IL method using SSL to improve the sample efficiency of expert demonstrations.

3.2 Self-supervised Representation Learning

Currently, SSL is divided into three approaches. The first is a pretext task [33], 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 [9]. The contrastive loss contrasts the neighboring instances with non-neighboring ones [11, 21]. 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 [17] and Simsiam [12] used Siamese networks with weight sharing and stop-gradient techniques to avoid collapse. Barlow twins [53] utilized a correlation matrix between the representations of a differently transformed same input to maximize the similarity between them while minimizing redundancy. VICReg [6] 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 [52] and SCARF [4], 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.

3.3 Inverse Reinforcement Learning

Although IRL [1, 56] 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

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 [7] 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	0.756	**0.843**	**0.843**
Predicted local outliers (%) ↓	90%	**6%**	26%	11%

overcome this problem. GAIL [22] is the first study drawing an analogy between IL and generative adversarial networks [16]. AIRL [15] proposed an AIL method that is robust to changes in dynamics. VAIL [35] improves the stability problem by constraining the information flow in the discriminator. EAIRL [38] 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 [47] covers the necessity of studying the sample efficiency of expert demonstrations, and proposes a simple model-based algorithm. Recently, f-GAIL [55] showed that finding an appropriate discrepancy measure during training is better than using a predetermined measure to improve sample efficiency. ASAF [5] 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 [23] or multiple experts [13]. 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 [50, 54]. 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.

4 Experiments

We assessed the performance of the proposed method on five continuous control benchmarks simulated by MuJoCo [46] (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:

$$\begin{aligned} \rho(s,a) &= \psi \rho_O(s,a) + \sum_{i=1}^{n} v_i \rho_i(s,a) \\ &= \psi \rho_O(s,a) + (1-\psi)\rho_N(s,a), \end{aligned} \tag{10}$$

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 $\psi + \sum_{i=1}^{n} v_i = 1$;

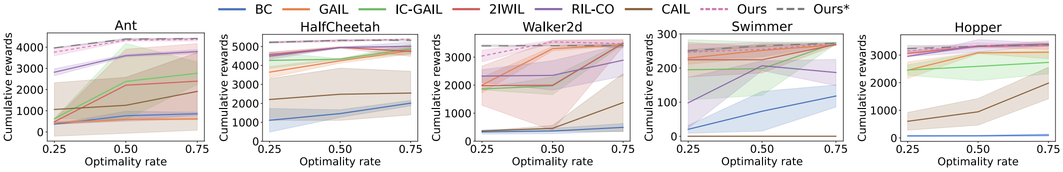

Figure 2: Final performance on five continuous control benchmarks with different optimality rates ψ. Vertical axes denote cumulative rewards acquired during the last 1000 training iterations. Shaded regions denote standard errors over three runs. Ours* = Ours + MM

that is, an optimality of 25% indicates that $\psi = 0.25$. More details on the experimental setting and hyperparameters can be found in the Supplementary S1.

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 S5 in the Supplementary S4, 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 [2]. 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 [8] (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 the Supplementary S8. Moreover, the average cumulative rewards of the expert policy that we obtained can be found in the Supplementary S1.1.

Corruption method We verified 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 $\sigma^2 : \frac{1}{1000}\sum_{i=1}^{1000}\left(\sum_{j=1}^{dim(s)}(s'_{i,j} - \bar{s}'_{i,j})^2\right)$ 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 [7] 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 is imperative 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 [50]. 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

Table 3: Final performance using 25 optimal and 75 non-optimal state-action pairs (ψ = 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	**3436.0±15.8**
	✓	3764.6±241.2	5217.4±29.7	3039.6±295.6
✓	✓	**3966.0±26.2**	**5221.0±75.1**	3403.0±52.5

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		
FD	CR	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**

refer to the Supplementary S1.2 for detailed explanations about 2IWIL, other comparison methods, and the collected imperfect demonstrations. The pseudo-code of the combined algorithm can be found in the Supplementary S3.

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 [54] 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) [48], to the combined method. In previous studies [28, 29], it is shown that MM in the feature space enriched by SSL is further effective to improve performance. Through this experiment, we would like 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.

5 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. [45] 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.

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. In addition, to demonstrate the effectiveness of Gaussian noise, we conducted an experiment of not appending noise dimensions and observed a decrease on benchmarks, as provided in the Supplementary S5. 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 5 presents that SimCLR's performance is superior to other methods by a significant margin.

Table 5: Ablation studies using 50 optimal and 50 non-optimal state-action pairs on Ant-v2 to test the role of a loss function of $\mathcal{L}_F$.

	MSE	BYOL	Barlow twins	SimCLR
$\mathcal{L}_F$	2793.4± 98.5	752.1±51.0	3602.7±1164.8	**4384.7±49.2**

Loss function of $\mathcal{L}_{SC}$ and $\mathcal{L}_{AC}$ 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. For $\mathcal{L}_{AC}$, the Barlow twins and SimSiam losses showed the first- and second-best performance on average, respectively. Detailed analysis and intuitions of SSL loss functions for a state and action are provided in the Supplementary S6.

Sensitivity to corruption rate Recently, studies on the importance of the degree of transformation have been proposed in SSL. Jing et al. [24] showed that a strong transformation can cause dimensional collapse. To find an appropriate corruption rate, we studied the sensitivity to the corruption rate of state and action and reported the results in Table S8 in the Supplementary S7.

6 Conclusion

The sample efficiency of expert demonstrations is desirable in IL because obtaining a large number of expert demonstrations is often costly. Motivated by successes in SSL, we propose a sample-efficient IL 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.

In spite of the excellence 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 SSL losses are added.

Acknowledgments

This work was conducted by Center for Applied Research in Artificial Intelligence (CARAI) grant funded by Defense Acquisition Program Administration (DAPA) and Agency for Defense Development (ADD) (UD190031RD) and was supported in part by a research grant from Samsung SDI, Co., Ltd.

References

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

[2] R. Agarwal, M. Schwarzer, P. S. Castro, A. C. Courville, and M. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in Neural Information Processing Systems*, 34, 2021.

[3] S. Arora and P. Doshi. A survey of inverse reinforcement learning: Challenges, methods and progress. *Artificial Intelligence*, page 103500, 2021.

[4] D. Bahri, H. Jiang, Y. Tay, and D. Metzler. Scarf: Self-supervised contrastive learning using random feature corruption. *arXiv preprint arXiv:2106.15147*, 2021.

[5] P. Barde, J. Roy, W. Jeon, J. Pineau, C. Pal, and D. Nowrouzezahrai. Adversarial soft advantage fitting: Imitation learning without policy optimization. In *NeurIPS*, 2020.

[6] A. Bardes, J. Ponce, and Y. LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning. *arXiv preprint arXiv:2105.04906*, 2021.

[7] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.

[8] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[9] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

[10] M. Chen, Y. Wang, T. Liu, Z. Yang, X. Li, Z. Wang, and T. Zhao. On computation and generalization of generative adversarial imitation learning. In *International Conference on Learning Representations*, 2019.

[11] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

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

[13] C.-A. Cheng, A. Kolobov, and A. Agarwal. Policy improvement via imitation of multiple oracles. *Advances in Neural Information Processing Systems*, 33, 2020.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

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

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

[17] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Pires, Z. Guo, M. Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *Neural Information Processing Systems*, 2020.

[18] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2019.

[19] N. Hansen, R. Jangir, Y. Sun, G. Alenyà, P. Abbeel, A. A. Efros, L. Pinto, and X. Wang. Self-supervised policy adaptation during deployment. In *International Conference on Learning Representations*, 2020.

[20] N. Hansen, H. Su, and X. Wang. Stabilizing deep q-learning with convnets and vision transformers under data augmentation. *Advances in Neural Information Processing Systems*, 34, 2021.

[21] K. He, H. Fan, Y. Wu, S. Xie, and R. 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.

[22] J. Ho and S. Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29:4565–4573, 2016.

[23] Z. Hu, Y. Liang, J. Zhang, Z. Li, and Y. Liu. Inference aided reinforcement learning for incentive mechanism design in crowdsourcing. *Advances in Neural Information Processing Systems*, 31:5507–5517, 2018.

[24] L. Jing, P. Vincent, Y. LeCun, and Y. Tian. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348*, 2021.

[25] K. P. Kielak. Do recent advancements in model-based deep reinforcement learning really improve data efficiency?, 2020.

[26] M. Laskin, K. Lee, A. Stooke, L. Pinto, P. Abbeel, and A. Srinivas. Reinforcement learning with augmented data. *Advances in Neural Information Processing Systems*, 33, 2020.

[27] A. Lee, A. Nagabandi, P. Abbeel, and S. Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *Advances in Neural Information Processing Systems*, 33, 2020.

[28] K. Lee, Y. Zhu, K. Sohn, C.-L. Li, J. Shin, and H. Lee. i-mix: A domain-agnostic strategy for contrastive representation learning. 2021.

[29] P. Mangla, N. Kumari, A. Sinha, M. Singh, B. Krishnamurthy, and V. 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.

[30] B. Mazoure, R. Tachet des Combes, T. L. DOAN, P. Bachman, and R. D. Hjelm. Deep reinforcement and infomax learning. *Advances in Neural Information Processing Systems*, 33, 2020.

[31] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[32] S. Mohamed and D. Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. *Advances in Neural Information Processing Systems*, 28:2125–2133, 2015.

[33] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, 2016.

[34] D. Pathak, P. Agrawal, A. A. Efros, and T. Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.

[35] X. B. Peng, A. Kanazawa, S. Toyer, P. Abbeel, and S. Levine. Variational discriminator bottleneck: Improving imitation learning, inverse RL, and GANs by constraining information flow. In *International Conference on Learning Representations*, 2019.

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

[37] B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, and G. Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180. PMLR, 2019.

[38] A. H. Qureshi, B. Boots, and M. C. Yip. Adversarial imitation via variational inverse reinforcement learning. In *International Conference on Learning Representations*, 2018.

[39] S. Reddy, A. D. Dragan, and S. Levine. Sqil: Imitation learning via reinforcement learning with sparse rewards. In *International Conference on Learning Representations*, 2019.

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

[41] M. Schwarzer, A. Anand, R. Goel, R. D. Hjelm, A. Courville, and P. Bachman. Data-efficient reinforcement learning with self-predictive representations. In *International Conference on Learning Representations*, 2020.

[42] R. Simmons-Edler, B. Eisner, D. Yang, A. Bisulco, E. Mitchell, S. Seung, and D. Lee. Reward prediction error as an exploration objective in deep rl. *arXiv preprint arXiv:1906.08189*, 2019.

[43] A. Srinivas, M. Laskin, and P. Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020.

[44] R. S. Sutton and A. G. Barto. Introduction to reinforcement learning, 1998.

[45] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola. What makes for good views for contrastive learning? *arXiv preprint arXiv:2005.10243*, 2020.

[46] E. Todorov, T. Erez, and Y. 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.

[47] F. Torabi, S. Geiger, G. Warnell, and P. Stone. Sample-efficient adversarial imitation learning from observation. *arXiv preprint arXiv:1906.07374*, 2019.

[48] V. Verma, A. Lamb, C. Beckham, A. Najafi, I. Mitliagkas, D. Lopez-Paz, and Y. Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR, 2019.

[49] T. Wang and P. Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.

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

[51] D. Yarats, I. Kostrikov, and R. Fergus. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International Conference on Learning Representations*, 2020.

[52] J. Yoon, Y. Zhang, J. Jordon, and M. van der Schaar. Vime: Extending the success of self-and semi-supervised learning to tabular domain. *Advances in Neural Information Processing Systems*, 33, 2020.

[53] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. Barlow twins: Self-supervised learning via redundancy reduction. In M. Meila and T. 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.

[54] S. Zhang, Z. Cao, D. Sadigh, and Y. Sui. Confidence-aware imitation learning from demonstrations with varying optimality. In *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[55] X. Zhang, Y. Li, Z. Zhang, and Z.-L. Zhang. f-gail: Learning f-divergence for generative adversarial imitation learning. *Advances in neural information processing systems*, 2020.

[56] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd national conference on Artificial intelligence-Volume 3*, pages 1433–1438, 2008.