Sample Efficient Imitation Learning via Reward Function Trained in Advance

Lihua Zhang$^1$, Quan Liu$^{1,2,3}$

$^1$School of Computer Science and Technology, Soochow University, Soochow
$^2$Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Soochow
$^3$Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Jilin

lihua_zhang92@163.com, quanliu@suda.edu.cn

Abstract

Imitation learning (IL) is a framework that learns to imitate expert behavior from demonstrations. Recently, IL shows promising results on high dimensional and control tasks. However, IL typically suffers from sample inefficiency in terms of environment interaction, which severely limits their application to simulated domains. In industrial applications, learner usually have a high interaction cost, the more interactions with environment, the more damage it causes to the environment and the learner itself. In this article, we make an effort to improve sample efficiency by introducing a novel scheme of inverse reinforcement learning. Our method, which we call Model Reward Function Based Imitation Learning (MRFIL), uses an ensemble dynamic model as a reward function, what is trained with expert demonstrations. The key idea is to provide the agent with an incentive to match the demonstrations over a long horizon, by providing a positive reward upon encountering states in line with the expert demonstration distribution. In addition, we demonstrate the convergence guarantee for new objective function. Experimental results show that our algorithm reaches the competitive performance and significantly reducing the environment interactions compared to IL methods.

1 Introduction

Imitation Learning (IL), also known as Learning from Demonstration (LfD), is a framework which learning an optimal behavior policy from expert demonstrations. The expert demonstration generates by expert policy, and composed of state-action pair. Each state-action pair indicates the action to take at the state being visited. There are two ways to attaining the goal that learning optimal policy from demonstrations, include Behavioral Cloning (BC) (Bain and Sammut 1995) and Inverse Reinforcement Learning (IRL) (Ng and Russell 2000).

BC methods learn an expert policy in a supervised fashion without environment interactions. Usually, behavioral cloning methods are the first option when have sufficient expert demonstrations (Sasaki, Yohira, and Kawaguchi 2019). In the practical application, such as autonomous vehicles task, the environment usually has a vast majority of state-action pairs, however, the expert demonstrations only have a limited number of the states. Therefore, BC methods often suffer from compounding error (Ross and Bagnell 2010), inaccuracies compound over time and can lead the learner to encounter unseen states in the expert demonstrations. Moreover, BC often can’t take the optimal action when encounters unseen states.

Inverse Reinforcement Learning (IRL) algorithm provide for an automated framework for decision making and control without reward function: by specifying a high-lever objective function, an IRL algorithm can, in principle, automatically learn a reward function and control policy that satisfies this objective. This has the potential to automate a range of applications, such as autonomous vehicles and robotic control. Given that the policy learns based on reward function, is the most succinct and robust way. Inverse Reinforcement Learning methods aim to recover reward or cost function by expert demonstrations, then IRL methods obtains optimal policy through a standard reinforcement learning algorithm. The target of the objective function is to train a reward function which guarantees the optimal policy trained by it as optimal as expert policy. IRL methods can take better action compared with BC algorithms, when encounters unseen states in the expert demonstrations. There for IRL can overcome the compounding error problem (Ziebart et al. 2008).

The original IRL framework utilizes a linear mapping between input features and out reward. These algorithms assume the unknown reward function that can be expressed as a linear combination of state-action pair’s features, this assumption severely restricts the complexity of reward structures that can be modeled accurately, and the reward function is difficult to apply to complex, high-dimensional tasks with large state-action space.

Finn and Levine who proposed an algorithm that use expressive, nonlinear function approximators, such as neural networks, to represent the reward function (Finn, Levine, and Abbeel 2016). This approach learns nonlinear reward function from expert demonstrations, at the same time, the approach learns a policy to perform the task, and the policy optimization “guides” the reward toward good regions of the space.

To improve the representation ability of linear reward function, Levine proposed Gaussian Process Inverse Reinforcement Learning (GP-IRL) (Levine, Popovic, and Koltun 2011) that recovers the reward function with Gaussian Process (GP). GP-IRL extends the GP model to account for the stochastic relationship between actions and underlying re-
wards. This allows GPIRL to balance the simplicity of the learned reward function against its consistency with the expert’s actions, without assuming the expert to be optimal. But GPIRL have a runtime dependent on the size of the dataset and are hard to processing large amounts of demonstration data.

There has some recent work on IRL algorithms that focus on recover reward function efficiently. Ho and Ermon who introduced a model-free imitation learning method called Generative adversarial Imitation Learning (GAIL) (Ho and Ermon 2016). This algorithm intimately connected to generative adversarial network (Goodfellow et al. 2014), and does not interact with expert during training. The process of training generator and discriminator network is divided into two procedures: RL procedure and IRL procedure. IRL is a dual of an occupancy measure matching problem, in this procedure, discriminator manage to improve its ability to distinguish which action bring a policy’s occupancy measure closer to the expert’s. Unlike DAgger (Ross, Gordon, and Bagnell 2011), can simply ask the expert for such actions.

GAIL, suffers from the problems of mode collapse and low sample efficiency in terms of environment interaction (Le et al. 2019). The weakness of mode collapse is inheriting from GANs, and several works have built on GAIL to overcome this problem (Li, Song, and Ermon 2017) (Fei et al. 2020). Although, GAIL leverage the expert demonstrations efficiently, it needs a large number of environment interactions. Similar to GAIL, other model free inverse reinforcement learning algorithms are quite data-expensive to train, which often limits their application to simulated domains. These methods require solving the reinforcement learning procedure (finding an optimal policy given the current reward function) in the inner loop of an iterative reward optimization. This makes them need a majority of interaction with environment when apply to continuous control task with large state-action space, where the reinforcement learning procedure is difficult to train. Particularly real-world systems with unknown dynamics.

We are interested in imitation learning and want to address this issue, because we desire an algorithm that can be applied to real-word problems for which it is hard to design the reward, and the cost of interaction with environment is expensive. Furthermore, in most real-word problems, even if the expert safely demonstrated, the learner may have policies that damage the environments and the learner itself during training. In this paper we focus on model-free imitation learning for continuous control. To address this problem, we propose a new approach to improve sample efficiency suffered by previous methods, which we call Model Reward Function Based Imitation Learning (MRFIL). The main contribution of our work is we dividing IRL algorithm into two steps: IRL procedure and RL procedure. In IRL procedure, MRFIL pre-trains a fixed reward function by expert demonstrations. In RL procedure, based on the pre-trained reward function, MRFIL learns an optimal policy with standard RL algorithms. We also propose a new objective function, and analyze the convergence theoretically. Unlike prior IRL methods, by pre-training a fixed reward function and learning an optimal policy in a single RL procedure, we considerably shrink the amount of interactions that interact with environment. Our evaluation demonstrates the performance of our method on a set of simulated benchmark tasks, showing that it achieves state-of-the-art performance as compared with a number of imitation learning task what difficult to interact to environment.

2 Related Work

Currently, the problem of learning an optimal policy in sample efficient way is still not well understood. Sample efficiency imitation learning data to at least work of Guided Cost Learning (GCL), which updates the cost function in the inner loop of policy search. Specifically, GCL directly optimizing a trajectory distribution with respect to the current cost function using sample-efficient reinforcement learning algorithm. Thus, the cost function can lead the learner toward regions where the samples are more useful, and inherits its sample efficiency. However, GCL require the model is well-approximated by iteratively fitted time-varying linear dynamics, and have worse imitation capability compared with GAIL (Fu, Luo, and Levine 2017).

GAIL is based on prior IRL works, and has achieved state-of-the-art performance on a variety of continuous control tasks, GAIL overcome the compounding error of BC methods, and is quite sample efficient than BC in terms of expert demonstration data. However, it suffers from the sample efficient in terms of environment interaction, which severe restricts its application in practical problems.

Hester and Osband proposed an approach use a prioritized replay mechanism to automate learn optimal policy (Hester et al. 2018). However, this algorithm needs expert demonstration and hand-crafted reward, we address the problem only need expert demonstration.

At present, the adversarial IL (AIL) framework has become a popular choice for IL (Baram et al. 2017); (Hausman et al. 2017); (Li, Song, and Ermon 2017). But these methods require a majority of state-action pairs obtained through the interaction between the learner and environment. To address this problem, Blondé and Kalousis leveraging an off-policy architecture to reduce interaction number with environment (Blondé and Kalousis 2019). However, this algorithm still needs reward function to learn through interact with environment.

Fumihiro and Tetsuya adopt off-policy actor-critic (Off-PAC) algorithm (Sasaki, Yohira, and Kawaguchi 2019). Compare this algorithm estimating the state-action value using off-policy samples without learning reward function. In contrast, our method trains reward function with offline.

3 Background

3.1 Markov Decision Process

We focus on model-free imitation learning for continuous control task in this work, and model this task as a Markov Decision Process (MDP), can be formalized by the tuple:$(S, A, p, r, \gamma, \rho_0)$. Where $S$ is the state space, $A$ is the action space, $p$ is the unknown transition function,
p : S × A × S → [0, 1], specifying the probability density p(s′, r|s, a) to the next state s′ acquire reward r from the current state s by taking action a. r : S × A × S → R is the reward function, ρ0 : S → [0, 1] is the distribution of initial states. A policy is a function π : S × A → [0, 1], which outputs a distribution over the action space for a given state s.

3.2 Policy Gradient Methods
In continuing tasks, where environment interactions are unbounded in sequence length, the returns R_t at time step t for a trajectory are defined as \( R_t = \sum_{k=t}^{\infty} \gamma^{k-t} r(s_k, a_k, s_{k+1}) \).

The policy gradient theorem assumes that every trajectory starts in some particular (non-random) state s_0, and s_0 leads learner to initial states according to distribution \( \rho_0 \). Then, the goal is to learn a policy that maximizes expected returns in state s_0.

\[
J(\theta) \triangleq v_{\pi_\theta}(s_0) = \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t, s_{t+1})
\]  

(1)

where \( v_{\pi_\theta}(s_0) \) denotes the value of state s_0 under policy \( \pi \).

Thus, the performance depends on both the action selections and the distribution of states in which those selection are made, and that both of these are affected by the parameter of policy network (Sutton and Barto 2018).

3.3 Inverse Reinforcement Learning Methods
A common assumption in IRL is that the demonstrator utilizes a Markov Decision Process for decision making. Normally, IRL algorithms assumes the expert policy is optimal. The goal of IRL is to recover the unknown reward function from expert demonstrations (Osada et al. 2018). Recover the reward function can be beneficial when the reward function is the most parsimonious way to describe the desired behavior.

In IRL methods, the reward function is designed for reward function which makes the current policy as optimal as expert policy. The current policy is update using standard reinforcement learning algorithms based on the current estimator of the reward function. By repeating this procedure, the optimal policy and reward function can be obtained.

4 Proposed Method
Currently, imitation learning methods obtains a policy as optimal as expert policy thorough at least millions interaction with environment. This severely limited imitation learning methods be widely utilized in the industrial field, due to the high interaction cost. Therefore, we introduce a noval imitation learning method that successfully addresses the impeding sample efficiency, in the number of interaction with environment.

4.1 Algorithm Formulation
It has various reasons that cause a low sample inefficiency. First, IL learner lacks of prior knowledge about environment, thus, all the information and knowledge that assist learner to take an action when visit a state can only be acquired by interact with environment. Secondly, based on current reward function, IRL methods alternate execute RL procedure and IRL procedure, this hugely increased the interaction number. Last, when human being learn a skill by imitating expert, the skill of human improves faster by learning a few expert demonstrations and samples. However, comparing with human, IL agent requires a majority of expert demonstrations and samples. One of the reason is that human being learn a new task with Prior-Knowledge. Based on the above analysis, in order to reduce the number of interact with environment, MRFIL performs three important modifications.

1. Training an ensemble of dynamic model and a single dynamic model with expert demonstration data, then we use the variance predicted by an ensemble of dynamic model which composed of five dynamic models as a reward. The motivation is that the prediction of ensemble dynamic model in expert demonstration state-action space tends to certainty, conversely, and the prediction of ensemble dynamic model far from expert demonstration state-action space tends to uncertainty. Consequently, according to this characteristic, the reward function could identify whether the visited state belong to expert demonstrations, and the reward function is acquired before algorithm interacting with environment, and the policy can be optimized through reinforcement learning method. Besides, a single dynamic model is used to pre-process neural network.

2. In order to reduce the number that agent interactions with environment, we use a single dynamic model and expert demonstration data to train algorithm in an offline way. We pre-train the algorithm with multi branch and short rollout, and the rollouts is divided to two ways: exploration rollout and exploitation rollout. This pretreatment provides a regularizing effect before policy learning, and is helpful to reduce training time.

3. Our theoretical result show that, MRFIL will cannot guarantee the convergence, if still adopt traditional objective function used in RL methods. To solve this problem, we add a supervised loss term to the objective function, and the supervised loss term is as same as BC loss. Furthermore, we demonstrate that the modified objective function guarantees the constriency in our approach.

4.2 Ensemble Reward Function
Generally, inverse reinforcement learning algorithms recover reward function with a single neural network, and learn it iteratively. Such as the popular choice for now: adversarial imitation learning.

We simplify this process to two single procedure: IRL procedure and RL procedure. In IRL procedure, MRFIL pre-trains a reward function. The reward function is defined as an ensemble of dynamic model, and the ensemble dynamic model is \( M = m_1, m_2, \ldots, m_n \). We use expert demonstrations to train the ensemble dynamic model via standard supervised learning method, each single dynamic model is only differ by the initial weights. In this way, the reward function can be acquired before the RL procedure start, and the optimal policy can be learned by standard reinforcement
learning method. The input of dynamic model is state-action pair \((s, a)\), the output is next state \(s'\). After the training procedure finished, if the current state-action pair belong to expert demonstrations, the variance of ensemble dynamics will be small. Conversely, if the current state-action pair far from expert state-action pair, the variance of ensemble dynamics will be large. Based on this principle, we define a hyper-parameter \(T_h\), to measure whether the input state-action pair belong to expert demonstrations. Concretely, the hyper-parameter is defined on the variance the output of ensemble dynamic models due to Pinsker’s inequality. The ensemble reward function is as follows:

\[
\text{var}(M(s, a)) = \text{var}(m_1(s, a), \cdots, m_n(s, a))
\]  

\[
r(s, a) = \begin{cases} 
1 & \text{if } \text{var}(M(s, a)) > T_h, \\
0 & \text{else}.
\end{cases}
\]

This modified reward function encourages the learner to explore in the state-action space close to expert demonstrations, and the learner is capable to go back when it far from the expert state-action space.

### 4.3 Pre-training with Dynamic Model

We adopt actor-critic frame to learn optimal policy in RL procedure. In the most of case, the algorithm converges finally. However, in the initial progress, the actor and critic network have a large variance which is derived from policy gradient methods that caused algorithm unstable and slower. Naturally, this problem leads to demand more samples to adjust learner policy. To address this problem, and improve sample efficiency, we pretrain the actor-critic network in an offline way. Same as ensemble dynamic model, we train a single dynamic model via expert demonstrations, which be used to predict next state, and the reward still provide by the ensemble dynamics.

The reason that caused a gap between sampling in true dynamic and dynamic model include two aspects: prediction error due to training data, and distribution shift due to the policy encountering states outside expert policy state-action space (Janner et al. 2019).

In current settings, the dynamic model is a local optimal model, only accurate when encountering the state-action space belong to expert demonstrations. Even the local optimal cannot be guaranteed when expert demonstration is insufficient.

Based on the above analysis, we propose an offline pre-train algorithm via the expert demonstrations and dynamic model. In order to reduce the compound error with the increase of rollout length, we train the algorithm with multi branch and short rollout. The algorithm starts several rollouts under the state distribution of expert demonstrations, and the rollouts is divided to two types: exploration rollout and exploitation rollout.

In the exploration rollout procedure, learner acts with stochastic noise, and the distance that agent runs is closes. The noisy action leads learner to “sub-optimal” region mostly, which is unfamiliar to dynamic model. In this region the dynamic model has high prediction error, but the ensemble dynamic models provide zero reward in all regions that outside expert demonstrations. Therefore, the prediction error has no impact on the accuracy of learner. Furthermore, this provides a regularization effect during policy learning by penalizing policy that visit “sub-optimal” region (Kidambi et al. 2020).

In the exploitation rollout procedure, learner act similar to standard model-based methods, but the distance that agent runs is far. This procedure in order to imitates expert policy in “optimal” region, and the dynamic model is familiar to this region thus has high prediction accuracy. We propose a corollary to measure the performance of pre-trained learner, and the theoretical proof is given. This corollary could measure the gap between expert return in real environment and learner return in dynamic model. The expert return in real environment can be sampled in expert demonstrations, and the theoretically analysis presented in section 5.1. The more details about how to pre-train the IL agent are presented in algorithm 1.

### 4.4 Imitation Learning with Ensemble Reward Function

In the RL procedure, we adopt Soft Actor-Critic (SAC) algorithm. If use the SAC objective function directly, the convergence of MRFL will not guaranteed, and the theoretical analyses is given in section 5.2. Therefore, we modify the objective function by adding a supervise term about actor loss, this modification is as follows:

\[
\theta_{t+1} = \theta_t + \alpha \nabla \theta_t \log \pi(a_t|s_t, \theta_t)(r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}, \omega) - Q(s_t, a_t, \omega) + \tau \nabla \theta_t \| \pi_E(a_t|s_t) - \pi(a_t|s_t, \theta_t) \|_2)
\]

where \(\theta_t\) denotes the parameter of actor network, \(r_{t+1}\) denotes the reward acquired by take action \(a_t\) in state \(s_t\), which is provided by ensemble dynamics \(M\). \(\alpha, \gamma, \tau\) denotes hyper-parameters.
Theorem B3. Imitation in policy. We theoretically demonstrate this modification in Algorithm 2 reach to expert state-action space quickly, and supervise loss term enables the imitation policy in line 6 of Algorithm 2. Adding this more details are presented in Algorithm 2. The return gap $\eta_e - \eta_m$ to measure the pre-training performance.

5.2 Convergence of The Algorithm

The objective function of inverse reinforcement learning algorithms is as follows:

$$\min_{\rho \in \mathcal{D}} -H(\rho) + \mathbb{E}_{\pi}[r(s, a)]$$ (6)

where $H(\pi) \triangleq \mathbb{E}_\pi[ - \log \pi(a | s)]$ is $\gamma$ discounted causal entropy of policy $\pi$, $\pi_E$ denotes expert policy.

**Theorem 1.** The dual of inverse reinforcement problem is as follows:

$$\min_{\rho \in \mathcal{D}} -H(\rho)$$

s.t. $\rho(s, a) = \rho_E(s, a) \quad \forall s \in S, a \in A$ (7)

**Proof.** See Appendix B, Theorem B.1. Where we denote occupancy measure $\rho_E : S \times A \to \mathbb{R}$ as $\rho_\pi(s, a) = \pi(a | s) \sum_{t=0}^{\infty} \gamma^t P(s_t = s | s, a)$. Theorem 1 implies that the goal of IRL is to find an optimal policy that the occupancy measure equals to expert policy. Following Theorem 1, that the learner taking action as same as expert in every state visitation can be guaranteed. The objective function of policy gradient methods with entropy term, but without supervise term is as follows:

$$\min_{\rho \in \mathcal{D}} -H(\rho)$$

s.t. $\rho(s, a) \geq 0 \quad \forall s \in S, a \in A$ (8)

**Theorem 2.** The dual problem of policy gradient methods objective function (Eq (7)) is as follows:

$$\min_{\rho \in \mathcal{D}} -H(\rho)$$

s.t. $\rho(s, a) \geq 0 \quad \forall s \in S, a \in A$ (9)

**Proof.** See Appendix B, Theorem B.2. Theorem 2 implies that uses SAC to learn optimal policy requiring the occupancy measure $\rho(s, a)$ greater than zero, which is already satisfied. Thus, the reward function that learns from expert demonstrations must be very precise, but this mathematical condition is hard to satisfy without interaction with environment. To address this problem, we modify the objective function Eq(8), the modified objective function is as follows:

$$\min_{\pi} -H(\pi) - \mathbb{E}_\pi[r(s, a)]$$

s.t. $\rho(s, a) \geq 0 \quad \forall s \in S, a \in A$ (10)

**Theorem 3.** The dual problem of MRFIL objective function (Eq (9)) is as follows:

$$\min_{\rho \in \mathcal{D}} -H(\rho)$$

s.t. $\rho(s, a) \geq 0 \quad \forall s \in S, a \in A$ (11)

$$\pi(s, a) = \pi_E(a | s) \quad \forall (s, a) \in D_E$$

The more details are presented in algorithm 2. Adding this supervise loss term enables the imitation policy in line 6 of Algorithm 2 reach to expert state-action space quickly, and learning around this optima region. Moreover, this modification guarantees the imitation policy converge to expert policy. We theoretically demonstrate this modification in section 5.2, and the convergence is proved by mathematical illation in Theorem B3.

5 Theoretical Analysis

This section provides the formal theoretical analysis of Model Reward Function Based Imitation Learning.

5.1 The Return Gap of Offline Pre-training

Under such a scheme, we propose a corollary to measure the performance of pre-trained IL learner, the maximal gap between IL learner and expert can be obtained by Corollary A.1. The corollary is as follows:

**Corollary A.1.** (multi branch and short rollout). Suppose the real dynamic transformation is $p_r(s' | s, a)$, the dynamic model transformation is $p_m(s' | s, a)$. The return gap $\eta_e - \eta_m$ is as follows:

$$|\eta_e - \eta_m| \leq \frac{\gamma (\epsilon m + \epsilon e)}{(1 - \gamma)} + \frac{\epsilon e}{1 - \gamma}$$ (5)

**Proof.** See Appendix A, Corollary A.1.

$\eta_e$ denotes the returns of the expert policy in real dynamic MDP, $\eta_m$ denotes the returns of the current policy in dynamic model MDP. $\gamma$ denotes the hyperparameter, $\pi_e$ denotes expert policy, $\pi_m$ denotes the policy learned by pre-training. where $\max_{\pi} \mathbb{E}_\pi [r(s, a)] \leq \epsilon$, and $\max_{\pi} D_{KL}(p_r(s' | s, a) || p_m(s' | s, a)) \leq \epsilon$. The episode return is an indicator of learner performance, thus we return gap $\eta_e - \eta_m$ to measure the pre-training performance.

$$\max_{\pi} \left( \min_{r \in R} -H(\pi) - \mathbb{E}_\pi[r(s, a)] \right) + \mathbb{E}_{\pi_E}[r(s, a)]$$ (6)

Algorithm 2 Algorithm 2 MRFIL: Model Reward Function Based Imitation Learning for Sample Efficient

1: **Input:** Expert demonstration data $D_E = \{(S_E, A_E, S'_E)_i\}_{i=1}^n$
2: Initialize network weight $\theta = \{(S_E, A_E, S'_E), \pi, \pi_E, \gamma\}$
3: Initialize an empty replay pool $D_{re} = \emptyset$
4: Use BC method to pretrain $\pi_E, M = \{(S_E, A_E, S'_E)_i\}_{i=1}^n$ with data $D_E$
5: Use Algorithm 2 to pre-train the actor-critic network
6: while $\pi$ and $Q$ not converged do
7: Sample action $a_t$ from the policy $\pi$, and get transition $s_{t+1}$ from the environment
8: Get reward $r_{t+1}$ from ensemble dynamic model $D_{re} \leftarrow D_{re} \cup \{(s_t, a_t, r_{t+1}, s_{t+1})\}$ Sample from replay pool to minimize policy gradient and value gradient
9: end while
10: Output $\theta$

\[\theta_{t+1} = \theta_t + \alpha \nabla_{\theta_t} \log \pi(a_t | s_t, \theta_t) (r_{t+1}) + \gamma Q(s_{t+1}, a_{t+1}, \omega_t) - Q(s_t, a_t, \omega_t) + \gamma \nabla_{\theta_t} \log \pi_E(a_t | s_t) - \pi(a_t | s_t, \theta_t)\]
Figure 1: **Image-based MuJoCo**. Performance comparison between MRFIL, GAIL and SAMIL in terms of episodic return. The horizontal axis depicts, in logarithmic scale, the number of interactions with environment. Each algorithm is run across 5 random seeds.

**Proof.** See Appendix B, Theorem B.3.

In Theorem 3, the supervised term in Eq(10)) is converted to \( \pi(s, a) = \pi_E(a|s) \), this require the learner becoming as optimal as expert. The constraint in Theorem 3 play a role equal to the constraint in Theorem 1, thus the learner is guaranteed to take action as same as expert in every state visitation.

### 6 Experiments

In this section, we comparatively evaluate our proposed method on five continuous control tasks which is built with MuJoCo physics engine, and is wrapped via the OpenAI Gym API. We want to investigate two aspect of MRFIL: the effectiveness of learning from expert demonstrations, and the accuracy of reward function. Specifically, inorder to show its learning performance, we compare MRFIL to two algorithms: GAIL and Sample Efficient Imitation Learning (SAMIL) (Blondé and Kalousis 2019). The goal of this experiment is that investigate not only how well each method can mimic the expert demonstrations, but also how well MRFIL reduces the interaction numbers.

#### 6.1 Experimental Settings

All experiments in this section are conducted in a cluster with two machines with 2 NVIDIA Tesla P40 GPUs each. We implement our algorithms with PyTorch. The agent always starts from the origin point, MRFIL performs on all environments within a 1 million step threshold. We parameterize the actor-critic model using 2-layer ReLU-MLPs and use an ensemble of 5 dynamics models to implement reward function as described in Section 4.3, we parameterize the dynamic model using a 4-layer ReLU-MLPs. For both environments, a Gaussian noise of \( N(0, 0.2) \) was added to the states to introduce stochasticity.

#### 6.2 Empirical Results

We generate the expert demonstration by SAC that based on Gaussian policy, the expert demonstrations which include 15000 episodes are collected starting from the initial state distribution, and 70 percent of it are used in training process. In the experiment, to simplify the training procedure, the supervised loss term in objective function is computed by calculating the difference between BC model and learner, the BC model is trained by expert demonstration. The experiment result demonstrates that this way is feasible.

Results of all methods are shown in Figure 1. In these environments we found that MRFIL achieves a higher performance than GAIL and SAMIL in Hopper-v2, Ant-v2 and HalfCheetah-v2. GAIL performs somewhat better than MRFIL on Walker2D-v2. For Ant-v2, Due to the high action dimension, none of GAIL and SAMIL are able to learn optimal policy. The result demonstrates that MRFIL requires significantly less environment interaction on both experimental.

Since our approach pre-train the actor-critic network in offline way, and add a supervised loss item. Therefor, these modifications are able to guide the learner to explore nearby expert policy state-action space. Besides, the reward function
punishes learner to visit unknown states, thereby providing a safeguard against distribution shift.

7 Conclusion
This paper introduced MRFIL, a model-free IL algorithm for continuous control task. The reward function we introduce is acquired by an ensemble of dynamic model from expert demonstrations in offline way. In the sense that the procedure learner interacts with environment is a single RL procedure, this hugely reduce the interaction number. In addition, we propose a model-based offline imitation learning method to pre-train the learner. In the RL procedure, we present a new objective function. Theoretically, we provide a method to measure the return gap between learner and expert when training with model-based IL method. Furthermore, we demonstrate the convergence guarantee for new objective function. Experiment results suggests that MRFIL consistently delivers comparable performance to GAIL and SAMIL, meanwhile MRFIL significantly reduces the environment interactions. In future work, we hope to extend MRFIL to robotic control task.

A Measure the Policy in Dynamic Model
In this section we introduce a corollary to measure the return gap between real expert policy and current policy in dynamic model.

Corollary A 1. (multi branch and short rollout). Suppose the real dynamic transformation is \( p_m \), the dynamic model transformation is \( p_m \).

\[
|\eta_e - \eta_m| \leq 2 \left( \frac{\gamma (\epsilon_m + \epsilon_\pi)}{(1 - \gamma)^2} + \frac{\epsilon_\pi}{1 - \gamma} \right) \\
\text{s.t. } (s, a) \in \{(s, a) \mid r_m(s, a) = 1\}
\]

Proof. Here, \( \eta_e \) denotes returns of expert policy \( \pi_e \) under real dynamic environment transformation \( p_e \), and \( \eta_m \) denotes returns of policy \( \pi_m \) learned by dynamic model \( p_m \) under dynamics transformation \( p_m \). \( p_e(s, a) \) denotes the state-action pair probability in expert policy \( \pi_e \), \( p_m(s, a) \) denotes the state-action pair probability in current policy \( \pi_m \). \( r(s, a) \) denotes the real reward function, \( r_m(s, a) \) denotes the ensemble reward function.

\[
|\eta_e - \eta_m| = \left| \sum_{s,a} (p_e(s, a) - p_m(s, a)) r_m(s, a) \right| \\
= \sum_{s,a} \sum_t \gamma^t (p_e^t(s, a) - p_m^t(s, a)) r_m(s, a) \\
= \sum_{s,a} \sum_t \gamma^t (p_e^t(s, a) - p_m^t(s, a)) r_m(s, a)
\]

According the reward function (Eq.(2) and Eq.(3)), the reward \( r(s, a) \) must be equal to 1 or 0. Thus we get:

\[
|\eta_e - \eta_m| = \left| \sum_{s,a} \sum_t \gamma^t (p_e^t(s, a) - p_m^t(s, a)) r_m(s, a) \right| \\
= \sum_{s,a} \sum_t \gamma^t |(p_e^t(s, a) - p_m^t(s, a))| r_m(s, a) \\
\text{s.t. } \{(s, a) \mid r_m(s, a) = 1\}
\]

Applying Lemma C.2, using \( \delta = \epsilon_m + \epsilon_\pi \) (via Lemma C.1), where \( \max_{p_e} \mathbb{E}_{s \sim p_m(s)} |D_{KL}(p_e(s, a) \| p_m(s, a))| \leq \epsilon_m \), and \( \max_s D_{TV}(\pi_E(a|s)), \text{max}_a D_{TV}(\pi_E(a|s)) \leq \epsilon_\pi \). Then:

\[
D_{TV}(p_e(s) \| p_m(s)) \leq t (\epsilon_m + \epsilon_\pi) \tag{15}
\]

And since we assume \( \max_s D_{TV}(\pi_E(a|s)|\pi_m(a|s)) \leq \epsilon_\pi \), we get:

\[
|\eta_e - \eta_m| = \left| \sum_{s,a} \gamma^t |(p_e^t(s, a) - p_m^t(s, a))| \right| \\
\leq 2 \sum_{s,a} \gamma^t (\epsilon_m + \epsilon_\pi) = 2 \left( \frac{\gamma (\epsilon_m + \epsilon_\pi)}{(1 - \gamma)^2} + \frac{\epsilon_\pi}{1 - \gamma} \right) \\
\text{s.t. } \{(s, a) \mid r_m(s, a) = 1\}
\]

B The Convergence of Objective Function
In this section we introduce three theorems to demonstrate the convergency of our algorithm.

B.1 Proof of Theorem 1
Theorem 1. The dual of inverse reinforcement problem is as follows:

\[
\min_{p \in \mathcal{P}} -H(p) \\
\text{s.t. } p(s, a) = \rho_E(s, a) \forall s \in S, a \in A \tag{17}
\]

Proof. We denote occupancy measure \( \rho : S \times A \to \mathbb{R} \) as \( \rho(s, a) = \pi(s|a) \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi) \).

\[
\min_{p \in \mathcal{P}} -H(p) \\
\text{s.t. } p(s, a) = \rho_E(s, a) \forall s \in S, a \in A \tag{18}
\]

Where we denote occupancy measure \( \rho : S \times A \to \mathbb{R} \) as \( \rho(s, a) = \pi(s|a) \sum_{t=0}^{\infty} \gamma^t P(s_t = s|\pi) \), we introduce generalized Lagrange function, and assume \( \max_{\mathcal{P}} \max_{\rho(s,a)} L_1(\rho, c) = \Psi_1^\dagger: \)

\[
\Psi_1^\dagger = -H(p) + \sum_{s,a} \alpha_{s,a} (\rho_E(s, a) - \rho(s, a)) \tag{19}
\]
Denote the dual variables $\alpha_{s,a}$ as reward $r(s,a)$, and assume $\max_{(s,a)} \min_{\rho(s,a)} L_1(\rho, c) = \Psi_1^2$.

$$\Psi_1^2 = -H(\rho) + \sum_{s,a} r(s,a) (\rho_E(s,a) - \rho(s,a))$$

$$= -H(\rho) + \sum_{s,a} r(s,a) \rho_E(s,a) - \sum_{s,a} r(s,a) \rho(s,a)$$

$$= -H(\rho) + \sum_{s,a} r(s,a) \pi_E(a|s) \sum_t \gamma^t p^t_E(s_t = s|\pi_E)$$

$$- \sum_{s,a} r(s,a) \pi(a|s) \sum_t \gamma^t p^t(s_t = s|\pi)$$

$$= -H(\rho) + \sum_{s,a} \pi_E(a|s) \sum_t \gamma^t p^t_E(s_t = s|\pi_E) r(s,a)$$

$$- \sum_{s,a} \pi(a|s) \sum_t \gamma^t p^t(s_t = s|\pi) r(s,a)$$

$$= -H(\rho) - \mathbb{E}_\pi [r(s,a)] + \mathbb{E}_{\pi_E} [r(s,a)]$$

(20)

Applying Lemma C.4, we get:

$$\max_{\rho(s,a)} \min_{\pi(s,a)} L_1(\rho, c) = -H(\rho) - \mathbb{E}_\pi [r(s,a)] + \mathbb{E}_{\pi_E} [r(s,a)]$$

$$= -H(\rho) - \mathbb{E}_\pi [r(s,a)] + \mathbb{E}_{\pi_E} [r(s,a)]$$

(21)

**B.2 Proof of Theorem 2**

**Theorem 2.** The dual problem of policy gradient methods objective function (Eq (8)) is as follows:

$$\min_{\rho \in D} \max_{\alpha, \beta} -H(\rho)$$

s.t. $\rho(s,a) \geq 0, \forall s \in S, a \in A$

$$\Psi_3 = \sum_{s,a} \alpha_{s,a} (\pi_E(a|s) - \pi(a|s))$$

Proof. We introduce generalized Lagrange function $\max_{\alpha, \beta} \min_{\rho(s,a)} L_3(\rho, r, \alpha) = \Psi_3^2$:

$$\Psi_3^2 = -H(\rho) - \sum_{s,a} \pi(a|s) \sum_t \gamma^t p^t(s_t = s|\pi)$$

$$+ \sum_{s,a} \alpha_{s,a} (\pi_E(a|s) - \pi(a|s))$$

$$= -H(\rho) - \sum_{s,a} \pi(a|s) \sum_t \gamma^t p^t(s_t = s|\pi)$$

$$+ \sum_{s,a} \alpha_{s,a} (\pi_E(a|s) - \pi(a|s))$$

$$= -H(\rho) - \mathbb{E}_\pi [r(s,a)]$$

(28)

When policy distribution $\pi(s,a)$ and $\pi_E(s,a)$ is very close, the dual parameters $\alpha_{s,a}$ have minimal impact on $L_3(\rho, r, \alpha)$, which can be ignored. We random sample state-action pair in expert demonstration, and take $\alpha_{s,a}$ as sample probability, and assume $\max_{\pi(s,a)} \min_{\rho(s,a)} L_3(\rho, r, \alpha) = \Psi_3$. Therefore:

$$\max_{\pi(s,a)} \min_{\rho(s,a)} L_3(\rho, r, \alpha) = -H(\rho) - \mathbb{E}_\pi [r(s,a)]$$

$$+ \sum_{(s,a) \in D_E} \alpha_{s,a} (\pi_E(s,a) - \pi(s,a))$$

$$\approx -H(\rho) - \mathbb{E}_\pi [r(s,a)]$$

$$+ \mathbb{E}_{D_E} [\pi_E(s,a) - \pi(s,a)]$$

(29)
B.4 Useful Lemmas

In this section we introduce two useful lemmas used in our corollary, which is provided in (Janner et al. 2019).

**Lemma C.1 (TVD of Joint Distributions.)** Suppose we have two distributions \( p_1(x, y) = p_1(x)p_1(y|x) \) and \( p_2(x, y) = p_2(x)p_2(y|x) \). We assume the total variation distance \( D_{TV}(p_1(x, y) \parallel p_2(x, y)) \) as \( \Phi \), we can bound the total variation distance of the joint as:

\[
\Phi \leq D_{TV}(p_1(x) \parallel p_2(x)) + \mathbb{E}_{x \sim p_1}[D_{TV}(p_1(y|x) \parallel p_2(y|x))]
\]

**Proof.**

\[
\Phi = \frac{1}{2} \sum_{x,y} |p_1(x, y) - p_2(x, y)|
\]

\[
= \frac{1}{2} \sum_{x,y} |p_1(x)p_1(y|x) - p_2(x)p_2(y|x)|
\]

\[
= \frac{1}{2} \sum_{x,y} (|p_1(x)p_1(y|x) - p_1(x)p_2(y|x)| + (p_1(x) - p_2(x))p_2(y|x))
\]

\[
\leq \frac{1}{2} \sum_{x,y} (p_1(x)p_1(y|x) - p_1(x)p_2(y|x))
\]

\[
\leq \frac{1}{2} \sum_{x,y} (|p_1(x)|p_1(y|x) - p_2(y|x))
\]

\[
= \mathbb{E}_{x \sim p_1}[D_{TV}(p_1(y|x) \parallel p_2(y|x))]
\]

\[
\leq D_{TV}(p_1(y|x) \parallel p_2(y|x))
\]

(30)

\[
\epsilon_t = D_{TV}(p^t_1(s) \parallel p^t_2(s))
\]

\[
= \frac{1}{2} \sum_s |p^t_1(s) - p^t_2(s)|
\]

\[
= \frac{1}{2} \sum_s (|p^t_1(s) - p^t_2(s)| + (p^t_1(s) - p^t_2(s)))
\]

\[
= \mathbb{E}_{s' \sim p^t_1}[|p_1(s|s') - p_2(s|s')|]
\]

\[
= \sum_{s'} \mathbb{E}_{s' \sim p^t_1}[|p_1(s|s') - p_2(s|s')|]
\]

(31)

\[
\delta_t = \epsilon_t - \epsilon_{t-1}
\]

\[
= \mathbb{E}_{s \sim p^t_1}[|p_1(s) - p_2(s)|]
\]

(32)

\[
\geq \mathbb{E}_{s \sim p^t_1}[|p_1(s) - p_2(s)|]
\]

(33)

Where we have defined \( \delta_t = \frac{1}{2} \sum_{s' \sim p^t_1} |\sum_{a \in A} p_1(s | a) - p_2(s | a')| \), which we assume is upper bounded by \( \delta \).

Assuming we are not modeling the initial state distribution, we can set \( \epsilon_0 = 0 \).

\[
\text{Lemma C.2 (Markov chain TVD bound, time-varying).} \text{ Suppose the expert KL-divergence between two transition distribution is bounded as and the initial state distribution are the same } p^t_1(s) = p^t_2(s). \text{ Then the distance in the state marginal is bounded as:}
\]

\[
D_{TV}(p^t_1(s) \parallel p^t_2(s)) \leq t\delta
\]

**Proof.** We begin by bounding the TVD in state-visitation at time \( t \), which is denoted as \( \epsilon_t = (p^t_1(s) \parallel p^t_2(s)) \), and assume

\[
|p^t_1(s) - p^t_2(s)| \text{ as } \Gamma:
\]

\[
\Gamma = \sum_{s'} |p^t_1(s = s'|s') - p^t_2(s = s'|s')|
\]

\[
\leq \sum_{s'} |p^t_1(s = s'|s') - p^t_2(s = s'|s')|
\]

\[
= \sum_{s'} \left[ |p^t_1(s) - p^t_2(s|s')| \right]
\]

(34)

\[
= \sum_{s'} p(s|s') \left[ |p^t_1(s) - p^t_2(s|s')| \right]
\]

(35)

\[
\text{Lemma C.3 (Theorem 2 of (Syed, Bowling, and Schapire 2008)).} \text{ If } \rho \in D, \text{ then } \rho \text{ is the occupancy measure for } \pi_{\rho}(a|s) \equiv \frac{\rho(s,a)}{\sum_a \rho(s,a')}, \text{ and } \pi_{\rho} \text{ is the only policy whose occupancy measure is } \rho.
\]

**Lemma C.4 (Lemma 3.2 of (Ho and Ermon 2016))** Let \( H(\rho) = -\sum_{s,a} \rho(s,a) \log (\rho(s,a)/\sum_{a'} \rho(s,a')) \). Then, \( H(\rho) \) is strictly concave, and for all \( \pi \in \Pi \) and \( \rho \in D \), we have \( H(\pi) = H(\rho_{\pi}) \) and \( H(\rho) = H(\pi_{\rho}) \).
References

Bain, M., and Sammut, C. 1995. A framework for behavioural cloning. In Proceedings of Machine Intelligence 15, 103–129. Oxford, UK: Oxford University Press.

Baram, N.; Anschel, O.; Caspi, I.; and Mannor, S. 2017. End-to-end differentiable adversarial imitation learning. In Proceedings of the 34th International Conference on Machine Learning, 390–399. Sydney, Australia: PMLR.

Blondé, L., and Kalousis, A. 2019. Sample-efficient imitation learning via generative adversarial nets. In Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics, 3138–3148. Naha, Japan: PMLR.

Fei, C.; Wang, B.; Zhuang, Y.; Zhang, Z.; Hao, J.; Zhang, H.; Ji, X.; and Liu, W. 2020. Triple-gail: A multi-modal imitation learning framework with generative adversarial nets. In Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, 2929–2935. Yokohama, Japan / Online: ijcai.org.

Finn, C.; Levine, S.; and Abbeel, P. 2016. Guided cost learning: Deep inverse optimal control via policy optimization. In Proceedings of the 33rd International Conference on Machine Learning, 49–58. New York City, USA: JMLR.org.

Fu, J.; Luo, K.; and Levine, S. 2017. Learning robust rewards with adversarial inverse reinforcement learning. CoRR abs/1710.11248(2):http://arxiv.org/abs/1710.11248.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A. C.; and Bengio, Y. 2014. Generative adversarial networks. In Proceedings of the advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems, 2672–2680. Montreal, Canada: Curran Associates Inc.

Hausman, K.; Chebotar, Y.; Schaal, S.; Sukhatme, G. S.; and Lim, J. J. 2017. Multi-modal imitation learning from unstructured demonstrations using generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 1235–1245. Long Beach, USA: Curran Associates Inc.

Hester, T.; Vecerik, M.; Pietquin, O.; Lanctot, M.; Schaul, T.; Piot, B.; Horgan, D.; Quan, J.; Sendonaris, A.; Osband, I.; Dulac-Arnold, G.; Agapiou, J. P.; Leibo, J. Z.; and Gruslys, A. 2018. Deep q-learning from demonstrations. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, 3223–3230. New Orleans, USA: AAAI Press.

Ho, J., and Ermon, S. 2016. Generative adversarial imitation learning. In Proceedings of the advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems, 4565–4573. Barcelona, Spain: Curran Associates Inc.

Janner, M.; Fu, J.; Zhang, M.; and Levine, S. 2019. When to trust your model: Model-based policy optimization. In Proceedings of the Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, 12498–12509. Vancouver, Canada: Curran Associates Inc.

Kidambi, R.; Rajeswaran, A.; Netrapalli, P.; and Joachims, T. 2020. Morel: Model-based offline reinforcement learning. In Proceedings of the Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems. Vancouver, Canada / Online: Morgan Kaufmann.

Le, T.; Hoang, Q.; Vu, H.; Nguyen, T. D.; Bui, H.; and Phung, D. Q. 2019. Learning generative adversarial networks from multiple data sources. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, 2823–2829. Macao, China: ijcai.org.

Levine, S.; Popovic, Z.; and Koltun, V. 2011. Nonlinear inverse reinforcement learning with gaussian processes. In Proceedings of the advances in neural information processing systems, 19–27. Granada, Spain: Curran Associates Inc.

Li, Y.; Song, J.; and Ermon, S. 2017. Infogail: Interpretable imitation learning from visual demonstrations. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems, 3812–3822. Long Beach, USA: Curran Associates Inc.

Ng, A. Y., and Russell, S. J. 2000. Algorithms for inverse reinforcement learning. In Proceedings of the Seventeenth International Conference on Machine Learning, 663–670. Stanford, USA: Morgan Kaufmann.

Osa, T.; Pajarinen, J.; Neumann, G.; Bagnell, J. A.; Abbeel, P.; and Peters, J. 2018. An algorithmic perspective on imitation learning. Found. Trends Robotics 7(1-2):1–179.

Ross, S., and Bagnell, D. 2010. Efficient reductions for imitation learning. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, 661–668. Sardinia, Italy: JMLR.org.

Ross, S.; Gordon, G. J.; and Bagnell, D. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 627–635. Fort Lauderdale, USA: JMLR.org.

Sasaki, F.; Yohira, T.; and Kawaguchi, A. 2019. Sample efficient imitation learning for continuous control. In Proceedings of the 7th International Conference on Learning Representations. L.A, USA: OpenReview.net.

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

Syed, U.; Bowling, M. H.; and Schapire, R. E. 2008. When to trust your model: Model-based policy optimization. In Proceedings of the Twenty-Fifth International Conference, 1032–1039. Helsinki, Finland: ACM.

Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; and Dey, A. K. 2008. Maximum entropy inverse reinforcement learning. In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, 1433–1438. Chicago, USA: AAAI Press.