MITSUBISHI ELECTRIC RESEARCH LABORATORIES
https://www.merl.com

Offline Imitation Learning upon Arbitrary Demonstrations by Pre-Training Dynamics Representations

Ma, Haitong; Dai, Bo; Ren, Zhaolin; Wang, Yebin; Li, Na

TR2025-147 October 23, 2025

Abstract

Limited data has become a major bottleneck in scaling up offline imitation learning (IL). In this paper, we propose enhancing IL performance under limited expert data by introducing a pre-training stage that learns dynamics representations, derived from factorizations of the transition dynamics. We first theoretically justify that the optimal decision variable of offline IL lies in the representation space, signifi- cantly reducing the parameters to learn in the downstream IL. Moreover, the dynamics representations can be learned from arbitrary data collected with the same dynamics, allowing the reuse of massive non-expert data and mitigating the limited data issues. We present a tractable loss function inspired by noise contrastive estimation to learn the dynamics representations at the pre-training stage. Experiments on MuJoCo demonstrate that our proposed algorithm can mimic expert policies with as few as a single trajectory. Experiments on real quadrupeds show that we can leverage pre-trained dynamics representations from simulator data to learn to walk from a few real-world demonstrations.

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) 2025

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

Offline Imitation Learning upon Arbitrary Demonstrations by Pre-Training Dynamics Representations

Haitong Ma, Bo Dai, Zhaolin Ren, Yebin Wang, Na Li

***Abstract*— Limited data has become a major bottleneck in scaling up offline imitation learning (IL). In this paper, we propose enhancing IL performance under limited expert data by introducing a pre-training stage that learns *dynamics representations*, derived from factorizations of the transition dynamics. We first theoretically justify that the optimal decision variable of offline IL lies in the representation space, significantly reducing the parameters to learn in the downstream IL. Moreover, the dynamics representations can be learned from arbitrary data collected with the same dynamics, allowing the reuse of massive non-expert data and mitigating the limited data issues. We present a tractable loss function inspired by noise contrastive estimation to learn the dynamics representations at the pre-training stage. Experiments on MuJoCo demonstrate that our proposed algorithm can mimic expert policies with as few as a single trajectory. Experiments on real quadrupeds show that we can leverage pre-trained dynamics representations from simulator data to learn to walk from a few real-world demonstrations.**

I. Introduction

Offline imitation learning (IL) agents aim to mimic an expert policy using only a fixed dataset of expert demonstrations, without interacting with the environment through a behavior policy. Since offline IL eliminates the cost and risks associated with trial-and-error exploration, it has been widely applied to various robot learning tasks, including manipulation [1], [2] and locomotion [3], [4].

One of the most commonly used offline IL algorithms is behavior cloning [5], which uses supervised learning to match the expert policy and behavior policy directly, ignoring the Markovian properties. Behavior cloning suffers from the notorious compound error, meaning that a small learning error in a single step will drive the agent into scarce or unseen trajectories in the expert dataset, in turn amplifying the learning error after several steps [6]. To avoid compound error, IL is formulated as distribution matching between the behavior and expert state-action densities, leveraging the Markovian structure of expert data to mitigate the compound error [7]. This approach has been further extended to the offline setting [8] using DIstribution Correlation Estimation (DICE, [9]) as computation tools.

Haitong Ma, Zhaolin Ren, Na Li are with School of Engineering and Applied Sciences, Harvard University. Bo Dai is with School of Computational Science and Engineering, Georgia Institute of Technology. Yebin Wang is with Mitsubishi Electric Research Laboratories (MERL) and Na Li is a visiting scholar at MERL. Email: {haitongma, zhaolinren}@g.harvard.edu, bodai@cc.gatech.edu, yebinwang@ieee.org, nali@seas.harvard.edu.

The work is supported under NSF AI Institute: 2112085, NSF CNS: 2003111, NSF ECCS: 2401390, 2401391.

The Github link for open-sourced code, videos, and full report is available at https://congharvard.github.io/repr-imitation-learning/.

The major challenges in solving distribution matching offline are twofold, *i.e.*, limited expert data and solving $\min - \max$ optimization. In the offline setting, the agent can not interact with the environment and can only learn from limited expert data, resulting in heavy overfitting and poor generalization. Current solutions to limited data are mixing expert and sub-optimal data by regularization or weighted sum [8], [10]–[12]. However, the mixture still requires sub-optimality of the auxiliary data. Another significant issue is the computational complexity of $\min$-$\max$ optimization [9], [13], as DICE inherently performs a primal-dual optimization. Specifically, with neural network parametrizations in practice, the $\min$-$\max$ optimization becomes highly unstable [14].

To handle these challenges, we propose the *dynamics representations* to further leverage the dynamics information to improve offline IL. Specifically, we define dynamics representations from the factorization of system dynamics and theoretically justify their abilities to represent the decision variables of offline IL optimization. Therefore, we can conduct offline IL on the representation space, mitigating the computational difficulties of solving $\min - \max$ with neural networks and reducing parameters to learn. Moreover, the dynamics representations can be learned from arbitrary demonstrations with the same dynamics, relieving the limited data issue by learning representations from a large amount of non-expert or even random data. We formulate a two-stage algorithm that learn dynamic representations from all data with shared dynamics in the pre-training stage and conducts downstream IL on representation space with expert data only in the main stage.

We validate the proposed algorithm through locomotion tasks in both the MuJoCo simulator and real quadrupeds. The MuJoCo experiments demonstrate that dynamics representations enable learning locomotion policies with as few as a single expert trajectory. Meanwhile, real-world quadruped experiments show that the agent can learn to walk using 1000 seconds of demonstration data collected from real hardware, built upon dynamics representations learned from simulators.

II. Related works

A. Imitation Learning via Distribution Matching

Distribution matching has online and offline variants depending on whether the agent can interact with the environment. **Online distribution matching** starts from inverse reinforcement learning (IRL), where the agent tries to recover the reward from expert trajectories [15]–[17]. However, we need another RL process on the IRL output to recover the expert policy. [7] explains how the reward function

learning can be bypassed to directly learn expert policy by distribution matching. The algorithm starts the adversarial IL family that adversarially trains a behavior policy to mimic the expert and a discriminator to discriminate behavior and expert trajectory [18], [19]. **Offline distribution matching** is more challenging since we cannot sample from the behavior distribution, and the discriminator training is impossible. [8] first makes the offline distribution matching possible using DICE to estimate the density ratio between behavior policy and offline dataset [9]. Then offline IL focuses on the limited expert demonstration issue with different choices of regularization terms [8], [10], [11], [20].

B. Representation Learning for IL

In reinforcement learning (RL), successor representation is popular for learning and transfer [21], [22], but it is defined over reward functions and not capable for IL. Some work re-parametrized policies using latent variable models to learn task-agnostic skills on the latent space to represent general knowledge [23], [24], but it is designed only for learning from experts. In control theory, the Koopman operator theory [25] tries to lift the problem to a higher dimensional space where it becomes a linear system, but learning such mappings is very difficult.

Representations defined by factorization of transition dynamics [26], [27] have recently gained significant interest due to their rich representational capacity and transferability. The representability of dynamics factorization has been justified by theoretical analyses [28], [29] as well as empirical studies on sim-to-real transfer learning [30]. In this paper, we show that these representations are also compatible with IL after adding noise contrastive design inspired by contrastive learning [28], [31].

III. Imitation Learning via Distribution Matching

A. Problem Formulation

We use Markov Decision Processes (MDPs), a standard sequential decision-making model for our IL task. The MDP can be described as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, \rho, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P(\cdot|s,a) : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition operator with $\Delta(\mathcal{S})$ as the family of distributions over $\mathcal{S}$, $\rho \in \Delta(\mathcal{S})$ is the initial distribution and $\gamma \in (0,1)$ is the discount factor.

The goal of offline IL is to find a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ that mimics the behavior of given expert demonstrations. In the offline IL setting, we cannot interact with the MDP environment to collect samples with the execution policy π, but only access a dataset of transitions sampled from the MDP, $\mathcal{D} = \{(s_i, a_i, s_i'), |\ (s,a) \sim q, s' \sim P(\cdot \mid s,a), i = 1,2,\ldots,N\}$, where q is the data distribution. A subset $\mathcal{D}^{\exp} = \{(s_i, a_i, s_i'), |\ (s,a) \sim d^{\exp}, s' \sim P(\cdot \mid s,a), i = 1,2,\ldots,N\} \subseteq \mathcal{D}$ is the expert demonstrations, and $d^{\exp}$ is the distribution of state-action pairs generated by the expert.

The IL can be completed by matching the state-action stationary distribution $d^{\pi}(s)$ generated by execution policy π

$$d^{\pi}(s) := (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \Pr(s_t = s \mid s_0 \sim \rho, a_t \sim \pi(s_t), s_{t+1} \sim P(\cdot \mid s_t, a_t), \forall t), \tag{1}$$

with the expert demonstration distributions $d^{\exp}$. We abuse the notation to denote the stationary discounted distribution on state-action pairs, $d^{\pi}(s,a) := d^{\pi}(s)\,\pi(a \mid s)$.

We formulate the distribution matching problem as minimizing the f-divergence between behavior state-action distribution d with expert demonstration distribution $d^{\exp}$ while satisfying the density constraints,

$$\begin{aligned} &\min_{\pi,d} D_f(d \| d^{\exp}) = \mathbb{E}_{(s,a)\sim d^{\exp}}\left[f\left(\frac{d(s,a)}{d^{\exp}(s,a)}\right)\right] \\ &\text{s.t. } d(s',a') = (1-\gamma)\rho(s')\,\pi(a'|s') + \gamma \mathcal{P}_*^{\pi} d(s',a'), \\ &\quad \forall (s',a') \in \mathcal{S}\times\mathcal{A}. \end{aligned} \tag{2}$$

where d is a general probability measure, f-divergence $D_f : \Delta(\mathcal{S}) \times \Delta(\mathcal{S}) \to \mathbb{R}^+$ is a class of functions that measures the difference between two probability distributions defined with a *convex* function $f : (0,\infty) \to \mathbb{R}, f(1) = 0$ and the $\mathcal{P}_*^{\pi}$ transpose policy transition operator

$$\mathcal{P}_*^{\pi} d(s',a') = \pi(a' \mid s') \iint P(s' \mid s,a) d(s,a) ds da. \tag{3}$$

The optimization (2) is difficult to solve as the optimization variable is a function $d(\cdot,\cdot)$ and there are infinite many constraints for each (s,a) pair. Moreover, in offline setting, the execution policy π is not able to interact with environments, increasing the optimization difficulty.

B. Solution via Primal-dual Optimization

We start by constructing and reformulation the Lagrangian of problem (2) with $Q(s,a)$ as the dual variable,

$$\begin{aligned} &\max_{\pi,d}\min_{Q} -D_f(d\|d^{\exp}) + \\ &\iint Q(s,a)\cdot((1-\gamma)\rho(s)\pi(a|s) + \gamma\mathcal{P}_*^{\pi} d(s,a) - d(s,a))\, ds da \end{aligned} \tag{4}$$

Noticing the problem (2) is convex-concave given policy π, we can transform the Lagrangian (4) to

$$\begin{aligned} &\max_{\pi}\min_{Q}\ (1-\gamma)\cdot\mathbb{E}_{(s_0,a_0)\sim\rho\times\pi}\left[Q(s_0,a_0)\right] + \\ &\mathbb{E}_{(s,a)\sim d^{\exp}}\left[\max_{\nu}\nu(s,a)\cdot(\gamma\mathcal{P}^{\pi}Q(s,a) - Q(s,a)) - f(\nu(s,a))\right] \end{aligned} \tag{5}$$

where we first reparametrize the primal variable $d(s,a)$ to the density ratio

$$\nu(s,a) := \frac{d(s,a)}{d^{\exp}(s,a)}$$

and $\mathcal{P}^{\pi}$ is the adjoint operator of $\mathcal{P}_*^{\pi}$ defined as

$$\mathcal{P}^{\pi}Q(s,a) = \iint P(s' \mid s,a)\pi(a' \mid s')Q(s,a)ds'da'.$$

. The detailed derivation is deferred to Appendix B in our online report [32] due to space limit.

Simplification of (5) via Fenchel duality. Fenchel conjugate, or convex conjugate, indicates that any convex functions f can be written as

$$f(x) = \max_{\zeta\in\mathbb{R}}\left[x\cdot\zeta - f^*(\zeta)\right]$$

for any $x \in (0,+\infty)$, where f^* is the Fenchel conjugate or convex conjugate of f. f^* is also a convex function. Then

we can reformulate the second expectation in (5) as

$$\begin{aligned}&\max_{\nu} \nu(s,a)\cdot(\gamma\cdot\mathcal{P}^{\pi}Q(s,a)-Q(s,a))-f\left(\nu(s,a)\right)\\ =&f^{*}\left(\gamma\cdot\mathcal{P}^{\pi}Q(s,a)-Q(s,a)\right)\end{aligned} \tag{6}$$

Therefore, we can eliminate one optimization function and reduce the optimization complexity, resulting in

$$\begin{aligned}=&\max_{\pi}\min_{Q}\ (1-\gamma)\cdot\mathbb{E}_{(s_0,a_0)\sim\rho\times\pi}\left[Q\left(s_0,a_0\right)\right]+\\ &\mathbb{E}_{(s,a)\sim d^{\exp}}\left[f_{*}\left(\gamma\mathcal{P}^{\pi}\left[Q(s,a)\right]-Q(s,a)\right)\right]\end{aligned} \tag{7}$$

Optimality conditions of ν and Q. Most importantly, the optimal solution of convex conjugate in (6) indicates the optimal primal ν^{*}_{Q} given dual variable Q have the following relation,

$$f'\left(\nu^{*}_{Q}(s,a)\right)=\gamma\mathcal{P}^{\pi}Q(s,a)-Q(s,a) \tag{8}$$

by computing the derivatives of LHS of (6) and asking it to be zero.

Moreover, when both primal variable ν and dual variable Q are solved to the saddle points ν^{*},Q^{*} given policy π, the constraints in (2) given π are satisfied due to the saddle point optimality conditions. Therefore, the primal variable $d(s,a)$ recover $d^{\pi}(s,a)$ given π in (1), indicating $\nu^{*}(s,a)=\frac{d^{\pi}(s,a)}{d^{\exp}(s,a)}$ and

$$f'(\nu^{*}(s,a))=f'\left(\frac{d^{\pi}(s,a)}{d^{\exp}(s,a)}\right)=\gamma\mathcal{P}^{\pi}Q^{*}(s,a)-Q^{*}(s,a) \tag{9}$$

by extension of (8).

Remark 1 (Interpretation of the dual variable Q.)**.** Equation (8) shares a similar formulation with the policy evaluation in RL, *i.e.*, the dual variable Q can be interpreted as the state-action value function with respect to the reward function $-f'\left(\nu^{*}_{Q}(s,a)\right)$. Moreover, when Q is solved to optimal Q^{*}, it can be interpreted as the Q-function of $-f'\left(\frac{d^{\pi}(s,a)}{d^{\exp}(s,a)}\right)$. For example, if we select KL divergence where $f(x)=x\log(x)$, the reward function will be $-\log\frac{d^{\pi}(s,a)}{d^{\exp}(s,a)}-1$, *i.e.*, the log density ratio. From now on, we will use KL divergence instead of general f-divergence, indicating the following relation (ignoring the constant shift),

$$Q^{*}(s,a)=-\log\left(\frac{d^{\pi}(s,a)}{d^{\exp}(s,a)}\right)+\gamma\mathcal{P}^{\pi}Q^{*}(s,a) \tag{10}$$

Remark 2 (Difficulty in solving the $\min-\max$ optimization (7))**.** Solving (7) in practice is difficult in both computational and statistical aspects. **Statistically**, the imitation learning heavily relies on the expert trajectories in $d^{\exp}(s,a)$, which is expensive to collect. **Computationally**, with neural network introduced for parametrization of $Q\left(s,a\right)$, $\nu\left(s,a\right)$, and $\pi\left(a|s\right)$ in (7), the $\min$-$\max$ optimization is notoriously difficult and very unstable, therefore, usually requires many tuning and tricks, for example, gradient penalty from the generative adversarial training [8], [33]. These two factors together render inferior performance and poor generalization.

IV. Dynamics Representations

In this section, we define the *dynamics representations* from the factorization of system dynamics. We show that the dynamics representations can significantly help IL since they can fully represent $Q(s,a)$ in (6), enabling us to constrain optimization in the representation space. Moreover, the dynamics representations can be learned from arbitrary data sharing the same transition dynamics, reducing the statistical dependency on expert data only.

A. Definitions

Definition 1 (Dynamics representations)**.** There exists representations $\phi:\mathcal{S}\times\mathcal{A}\to\mathbb{R}^{k}$ and $\mu:\mathcal{S}\to\mathbb{R}^{k}$ such that

$$P(s'\mid s,a)=\langle\phi(s,a),\mu(s')p_{n}(s')\rangle \tag{11}$$

where the $p_{n}\in\Delta(\mathcal{S})$ is a noise distribution that has full support on $\mathcal{S}$, and $\langle\cdot,\cdot\rangle$ is vector inner product. □

Remark 3 (Noise distributions p_n and connection to linear MDP [26], [27], [29].)**.** Our factorization is similar to the linear MDP literature in the RL theory community except for the additional noise term $p_{n}(s')$ inspired by [28]. Adding the extra noise term has two benefits: **a)** Aligning with the density ratio learning in the offline setting when setting p_n as $d^{\exp}$ shown in Section IV-B; **b)** Enabling tractable loss function to learn representations ϕ,μ shown in Section V-A.

Remark 4 (Transferability and connections to successor features [21].)**.** We emphasize that our dynamics representations are only relevant with dynamics P, which can be naturally transferred across data collected from different policies or tasks sharing the same dynamics. Another popular family of feature transfer leverages the successor features [21], [22] sharing the similar decomposition of $Q(s,a)=\langle\psi^{\pi}(s,a),w\rangle$, which is obtained from the factorization of reward functions $r=\langle\mathbf{r}(s,a),w\rangle$ and $\psi^{\pi}(s,a)=\mathbb{E}_{\pi}[\mathbf{r}(s,a)]$. Note that the representation ψ^{π} is relevant with policy π, constraining its reuse within similar tasks only. Our dynamics representations ϕ,μ are irrelevant to policy, indicating more general transferability. Moreover, for our IL problem, no reward functions are explicitly defined, making leveraging the successor representations not practical.

Some might question the existence of such factorizations. We show an example of stochastic control with known dynamics,

Example 1 (Fournier random feature for nonlinear control with Gaussian noise [34].)**.** We give an example of features for a known nonlinear control system with Gaussian noise taht is commonly seen in robotic control, *i.e.*

$$s'=g(s,a)+\epsilon,\quad \epsilon\sim\mathcal{N}(0,\sigma^{2}I_{d})$$

where $g:\mathcal{S}\times\mathcal{A}\to\mathcal{S}$ is a deterministic nonlinear dynamics, and ϵ is the Gaussian noise. We can regard the transition dynamics as $P(s'\mid s,a)\propto\exp(-\|g(s,a)-s'\|^{2}/2\sigma^{2})$, which is a Gaussian kernel. Then according to Bonchner theorem [35], we have the Fourier random features of the Gaussian kernel, $P(s'\mid s,a)=\langle\psi_{\omega}(g(s,a)),\psi_{\omega}\left(s'\right)\rangle_{\mathcal{N}(\omega)}$ where $\psi_{w}(x)=\exp(\mathrm{i}\omega^{\top}x)$ and $\langle\cdot,\cdot\rangle_{\mathcal{N}(\omega)}=\mathbb{E}_{\omega\sim\mathcal{N}(0,\sigma^{-2}I_{d})}\left[\langle\cdot,\cdot\rangle\right]$. It translate to infinite-dimensional representations ϕ,μ whose i^{th} element are

$$\phi_{i}(s,a)=\psi_{w_{i}}(g(s,a)),\mu_{i}(s')=\psi_{w_{i}}(s')/p_{n}(s')$$

, respectively, where $\omega_{i}\sim\mathcal{N}(0,\sigma^{-2}I_{d})$. In this case, ϕ,μ are both infinite-dimensional features, but we can use finite-

dimensional truncation as an approximation in practice with provable approximation guarantees [34].

B. Representational Capacity of Dynamics Representations

We show that the proposed dynamics representations ϕ, μ can fully represent the dual variable Q in (6). We first show what is the representation space of ϕ, μ, respectively.

Density ratio $\frac{d^\pi(s)}{p_n(s)}$ represented by $\mu(s)$. Recalling the recursion on $d^\pi(s)$ in (2), substituting the factorization (11) in it, and dividing both sides by $p_n(s')$, we have

$$\frac{d^\pi(s')}{p_n(s')} = (1-\gamma)\frac{\rho(s')}{p_n(s')} + \gamma \left\langle \mu(s'), \underbrace{\int \phi(s,a)\, d^\pi(s)\pi(a|s)\, dsda}_{\theta^\pi} \right\rangle. \tag{12}$$

where θ^π is a linear weight irrelevant with s'. As we know the initial distribution ρ and the noise distribution p_n, we have the full linear representations of $\mu(s)$, or the state density ratio $\frac{d^\pi(s)}{p_n(s)}$. Moreover, when setting $p_n = d^{\exp}$, the state-action density ratio can be further represented by

$$\begin{aligned} \nu^*(s,a) &= \frac{d(s)}{d^{\exp}(s)} \frac{\pi(a \mid s)}{\pi^{\exp}(a \mid s)} \\ &:= \left((1-\gamma)\frac{\rho(s)}{p_n(s)} + \gamma \langle \mu(s), \theta^\pi \rangle \right) \zeta(s,a) \end{aligned} \tag{13}$$

where $\zeta(s,a) := \frac{\pi(a|s)}{\pi^{\exp}(a|s)}$ is the policy ratio.

Dual variable Q represented by ϕ, μ jointly. From the optimal solution (9) and Remark 1 we observe that the optimal dual variable Q^* can be interpreted as the value function of reward $-\log(\nu^*(s,a))$. Substitute the dynamics representations (11) into (8),

$$\begin{aligned} &Q^*(s,a) = -\log \nu^*(s,a) + \gamma \mathcal{P}^\pi Q^*(s,a) \\ &= -\log \nu^*(s,a) + \\ &\left\langle \phi(s,a), \gamma \underbrace{\int \mu(s') p_n(s') \pi(a'|s') Q(s',a') ds' da'}_{\omega^\pi} \right\rangle \end{aligned} \tag{14}$$

Substitute the density factorization (13), we have the full representation of optimal dual variable Q^*,

$$\begin{aligned} &Q^*_{w^\pi, \theta^\pi}(s,a) = \\ &\underbrace{-\log \zeta(s,a)}_{\text{Offline computable}} + \underbrace{\phi^\top(s,a)\omega^\pi - \log\left(\mu(s)^\top \theta^\pi\right)}_{\text{Sit in the representation space}} + \underbrace{(1-\gamma)\frac{\rho(s)}{d^{\exp}(s)}}_{\text{Offline computable}} \end{aligned} \tag{15}$$

where $\omega^\pi \in \mathbb{R}^k$ is the linear weights on representations ϕ independent from s, a. Note that the policy ratio ζ and initial state density ratio $\frac{\rho(s)}{d^{\exp}(s)}$ are all offline computable from the expert dataset and current behavior policy. Therefore, we have the full parametrization structure of Q^*, where the parameters to optimize are only ω^π, θ^π, *i.e.*, the coefficients on representations ϕ, μ, respectively. In practice, we can directly put parameters ω, θ and optimize with gradient descent.

C. Imitation Learning with Dynamics Representations

As we know that the representational structure of optimal Q^*, we can constrain the optimization of dual variable Q within the representation space according to (15), we can substitute the representation of Q^* to the our objective function (7) and transferring optimizing Q to optimizing the parameters ω, θ, *i.e.*,

$$\begin{aligned} \max_\pi \min_{\omega,\theta} &\mathop{\mathbb{E}}_{(s,a)\sim d^{\exp}} \left[\exp\left(\gamma \mathcal{P}^\pi Q^*_{\omega,\theta}(s,a) - Q^*_{\omega,\theta}(s,a)\right) \right] \\ &+ (1-\gamma) \mathop{\mathbb{E}}_{(s_0,a_0)\sim \rho\times\pi} \left[Q^*_{\omega,\theta}(s_0,a_0) \right]. \end{aligned} \tag{16}$$

where $f_*(x) = \exp(x) - 1$ when we select KL divergence (constants in the objective functions are ignored). With the learned representation, we can constrain the `min` side of the optimization to the representation space, making it easier to solve `min-max` problems.

V. Two-stage Algorithm for Representation and Imitation Learning

In this section, we propose a two-stage algorithm and discuss practical considerations. We will first explain a tractable representation learning algorithm as the pre-training stage that admits arbitrary datasets sharing the same dynamics P, then discuss the main training stage conducting downstream IL on representation space.

A. Pre-training Stage: Dynamics Representation Learning

We show how to learn the representations ϕ, μ from data. We consider the case where we have a (small) expert dataset $\mathcal{D}^{\exp}$ and a (large) general dataset $\mathcal{D}$ containing all data generated from the same dynamics P. Then we define the following representation learning objective function,

$$\begin{aligned} \min_{\phi,\mu} J_{\text{repr}}(\phi,\mu) := &-2\mathbb{E}_{(s,a,s')\sim\mathcal{D}} \left[\phi(s,a)^\top \mu(s') \right] \\ &+ \mathbb{E}_{(s,a)\sim\mathcal{D}, s_n\sim\mathcal{D}^{\exp}} \left[(\phi(s,a)^\top \mu(s_n))^2 \right] \end{aligned} \tag{17}$$

We show that by minimizing (17) we can get representation satisfying factorizations in (1) since

$$J_{\text{repr}}(\phi,\mu) = \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[\int l_{\phi,\mu}(s,a,s') ds' \right] - C$$

$$\text{where } l_{\phi,\mu}(s,a,s') := \left(\frac{P(s' \mid s,a)}{\sqrt{p_n(s')}} - \phi(s,a)^\top \mu(s') \sqrt{p_n(s')} \right)^2 \tag{18}$$

C is a constant irrelevant with ϕ and μ. The detailed derivation can be found in Appendix B in our online report [32]. It is easy to see from (18) that the ϕ and μ minimizing J_{repr} satisfy the factorization in (11). To improve numerical stability, we add a log probability regularization term similar to [28],

$$\min_{\phi,\mu} J_{\text{repr}}(\phi,\mu) + \lambda_{\text{repr}} \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[\left(\log\left(\mathbb{E}_{s'} \left[\phi^\top(s,a)\mu(s') \right] \right) \right)^2 \right] \tag{19}$$

where $J_{\text{repr}}(\phi,\mu)$ is defined in (17), and λ_{repr} is the regularization weights.

B. Main Stage: IL on Representation Space

Equation (16) already showed the IL on representation space, we address some practical considerations here. In practice, the initial states can be sampled from $d^{\exp}$ without affecting the optimality (See discussion in Section 5.3 in [8]). Therefore, the $\frac{\rho(s)}{d^{\exp}(s)}$ equals constant 1. Moreover, to avoid

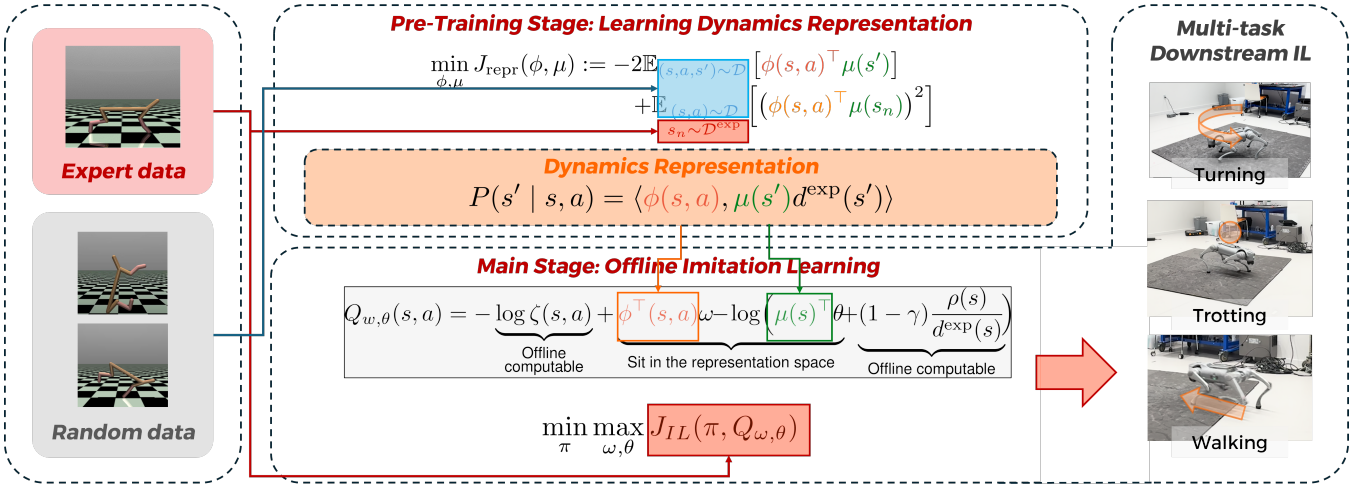

Fig. 1: Demonstration of the proposed algorithm. We first learn dynamics representations in the pre-training stage, and learn downstream IL in the main training stage within the representation space.

learning log policy ratio $\log \zeta(s,a)$ offline, we directly use a neural network $f_\xi(s,a)$ to fit it and arrives at practical Q parametrization,

$$Q_{\omega,\theta,\xi}(s,a) = f_\xi(s,a) + \phi(s,a)^\top \omega - \log\left(\mu(s)^\top \theta + 1 - \gamma\right) \tag{20}$$

TWe solve the inner dual variable estimation and outer policy optimization alternatively.

The overall two-stage IL algorithm is presented in Algorithm 1 as well as in Figure 1.

Algorithm 1 Representation ValueDICE (Abstract version)

Require: Expert trajectory dataset $\mathcal{D}^{\text{exp}}$, Dynamics dataset $\mathcal{D} \supseteq \mathcal{D}^{\text{exp}}$, initial policy π_0

1: `# representation learning stage`
2: Solve for representation ϕ, μ via solving (19) with dynamics dataset $\mathcal{D}$ and expert dataset $\mathcal{D}^{\text{exp}}$.
3: `# Imitation learning stage, alternatively do dual variable evaluation and policy update.`
4: **while** Training, alternatively **do**
5: Parameterize Q with learned representation ϕ, μ and parameters ω, θ, ξ following (20).
6: Optimize ω, θ, ξ parameters of dual variable Q via optimization (16).
7: Update policy π via optimization (16).
8: **end while**

VI. Experiments

The experimental tests aim to justify our claims on the representation-based DICE imitation learning, *i.e.*,

- *Does the representation improve IL performance and generalizations when expert data is limited?*
- *Can we reuse the representation from data sharing the same dynamics to help imitation learning?*

We will show results on locomotion tasks in MuJoCo simulators (with expert data only and a combination of expert and random data) and real quadrupeds.

A. Imitation Learning with Limited Expert Data

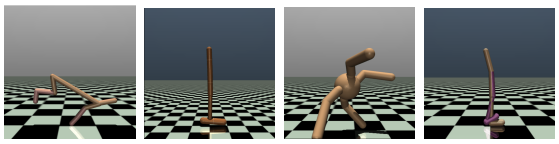

Fig. 2: Locomotion tasks in the MuJoCo simulator, from left to right: Half Cheetah, Hopper, Ant, and Walker2d.

Tasks and expert data. We conduct imitation learning on four locomotion tasks using MuJoCo physics simulator [36] shown in Figure 2, which are commonly used in previous IL papers [7], [8]. To answer question 1, we sample a limited number of expert trajectories from the expert dataset provided in [8], *i.e.*, **1 trajectory** for HalfCheetah, Hopper, and Walker2d, **3 trajectories** for Ant. The original ValueDICE paper [8] provides 40 trajectories. Each trajectory has a total of 1000 transitions.

Algorithms and Baselines. Other than the proposed algorithm **ReprValueDICE**, We include baselines (1) **ValueDICE** directly uses a NN to parameterize dual variable Q [8] and (2) behavior cloning (**BC**) that leverages maximum likelihood estimation to match expert and behaviour policies.

Evaluation and Performance. We evaluate the IL policy over 20 randomly initialized trajectories to assess the performance and demonstrate the generalization capabilities of policies learned from only 1 or 3 trajectories only, shown in Figure 3. The results show that ReprValueDICE achieves significantly better average returns than ValueDICE on HalfCheetah, Hopper, and Walker2d. For the Ant, ReprValueDICE shows stable performance consistently over the whole training process towards the expert, while BC and ValueDICE show good performance initially but then deteriorate quickly. All the results have shown that the dynamics representation can help mitigate overfitting and improve performance even with a little expert data, verifying that exploiting dynamics representation can improve the IL performance and generalization to unseen trajectories.

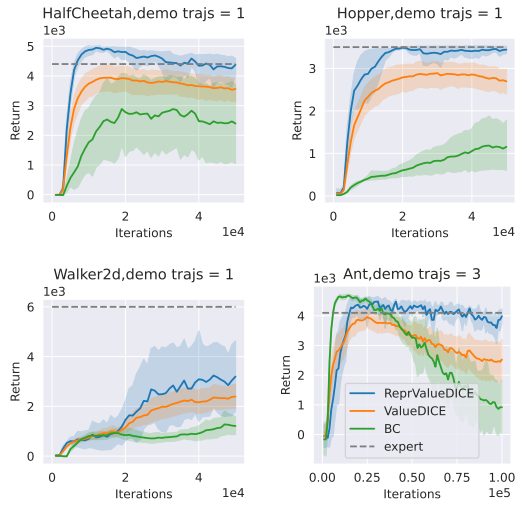

Fig. 3: Training performance of learning from limited expert data. Solid lines and shaded regions show the mean and standard deviation of episodic returns per 1000 training iterations with 10 random seeds.

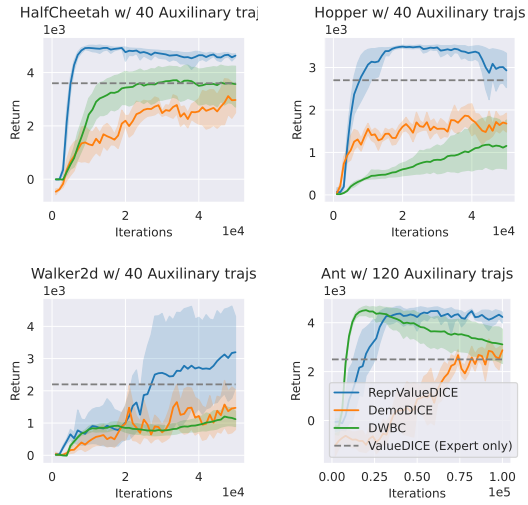

Fig. 4: Training performance of learning from expert data and auxiliary data. Solid lines and shaded regions show the mean and standard deviation of episodic returns per 1000 training iterations with 10 random seeds. Dashlines indicate learning the performance learning from expert data only using ValueDICE [8].

For sensitivities to other hyperparameters like the number of extra random data, the number of expert demonstrations, please refer to Appendix D in our online reports [32].

B. Imitation Learning with Auxiliary Data

Experimental Setup. We continue to conduct IL on MuJoCo locomotion tasks and leverage auxiliary data from D4RL [37] medium-level datasets to further help imitation learning. We consider appending 40 times more trajectories (120 trajectories for Ant and 40 trajectories for the other 3) of the previously mentioned mujoco tasks.

Algorithm and baselines. Our **ReprValueDICE** leverages the auxiliary data ane expert data together to learn representations in the pre-training stage and then imitates the expert data in the main stage. We compare our ReprValueDICE against a) **DemoDICE** [10] that uses the suboptimal data as regularization and 2) discriminator-weighted behavior cloning (**DWBC**) [18].

Evaluation and performance. We use the same evaluation process as Section VI-A. Experimental results have shown that ReprValueDICE shows significant improvements over ValueDICE with expert data only and outperforms DWBC and DemoDICE. Results on DWBC and DemoDICE show that when the medium-level data dominates the dataset, where in our case $\texttt{expert data:medium data}=1:40$, handling auxiliary data via regularization or weighted BC cannot achieve good performance.

C. Imitating Real Quadrupeds Controllers

We further show the superior expert data efficiency of proposed representation-based imitation learning on the real Unitree Go2 Quadrupeds trying to imitate the built-in controller on a task of walking, *i.e.*, tracking given base speed commands. Specifically, we consider three tasks, trotting (zero speed), walking ($0.5\mathrm{m/s}$ linear velocity) and turning ($0.5\ \mathrm{rad/s}$ angular velocity).

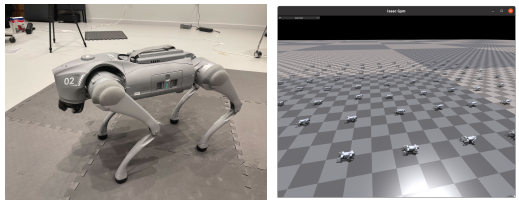

Fig. 5: We use the Unitree Go2 Quadrupeds and control it via the Jetson Orin NX extension deck (left). Auxiliary data are collected from the Issac Gym simulator (right).

Real-world and Simulator Data Collection. We collect 1000 seconds of real-world expert data using the built-in controller. Meanwhile, to show that we can learn and reuse representations from arbitrary data, we also use the Isaac Gym simulator to collect a large amount of auxiliary data. The data is collected during the process of training a reinforcement learning (RL) agent and is used as the dynamics dataset $\mathcal{D}$ to learn representations in equation (V-A).

Baselines and metrics. We compare the proposed ReprValueDICE with ValueDICE and BC that learns from expert data collected from real world only, showing that by leveraging dynamics representation, we can learn an effective walking controller with only a small amount of data. To get quantitative metrics on the gait behaviors, we compute the base walking height, foot air percentage, and average contact force partitions of front foot.

Results. We record the gait behavior metrics in Table I, which shows that the proposed ReprValueDICE well mimics the stock controller and demonstrates appropriate base height, less air time, and balance contact (average 25% on the front foot), all show a stable walking behavior. The RL always leans forward with a higher base, which easily leads to instability under external perturbation. As for ValueDICE and BC, they all fail to stably walk after learning from the same amount of data. Moreover, the velocities tracking performance of the real quadrupeds are shown in Appendix

D in our online report [32], showing the successful imitation of stock controllers. The video of learned policy are can also be found on our Project webpage.

TABLE I: Gait behaviour comparison averaging over all three speed tracking tasks. **The closer to the target stock controller in blue, the better the algorithm is**. Behavior cloning (BC) and ValueDICE from expert data only failed to walk stably.

	Base walking height (cm)	Foot air time percentage*(%)	Average front foot contact force partition
Stock (Target Policy)	32.4±2.2	24.0±3.3	24.8±12.2
ReprValueDICE	**31.6**±4.9	**25.0**±8.5	**27.6**±14.5
RL	37.8±5.3	30.0±11.3	42.4±14.5
BC	N/A	N/A	N/A
ValueDICE (from expert demo)	N/A	N/A	N/A

VII. Concluding Remarks

In this paper, we propose the dynamics representations to address the challenge of optimization and sample efficiency in offline IL. We define representations through a factorization of transition dynamics and show that it can fully represent the decision variable Q in offline IL. Experimental results on MuJoCo and real quadrupeds verified that the proposed algorithm has less overfitting and better generalization and can reuse auxiliary non-expert data to learn representations and improve algorithm performance.

References

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

[2] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," *arXiv preprint arXiv:2108.03298*, 2021.

[3] N. Ratliff, J. A. Bagnell, and S. S. Srinivasa, "Imitation learning for locomotion and manipulation," in *2007 7th IEEE-RAS international conference on humanoid robots*. IEEE, 2007, pp. 392–397.

[4] X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," *arXiv preprint arXiv:2004.00784*, 2020.

[5] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," *Advances in neural information processing systems*, vol. 1, 1988.

[6] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[7] J. Ho and S. Ermon, "Generative adversarial imitation learning," *Advances in neural information processing systems*, vol. 29, 2016.

[8] I. Kostrikov, O. Nachum, and J. Tompson, "Imitation learning via off-policy distribution matching," *arXiv preprint arXiv:1912.05032*, 2019.

[9] O. Nachum, Y. Chow, B. Dai, and L. Li, "Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections," *Advances in neural information processing systems*, vol. 32, 2019.

[10] G.-H. Kim, S. Seo, J. Lee, W. Jeon, H. Hwang, H. Yang, and K.-E. Kim, "Demodice: Offline imitation learning with supplementary imperfect demonstrations," in *International Conference on Learning Representations*, 2022.

[11] Y. Ma, A. Shen, D. Jayaraman, and O. Bastani, "Versatile offline imitation from observations and examples via regularized state-occupancy matching," in *International Conference on Machine Learning*. PMLR, 2022, pp. 14 639–14 663.

[12] F. Sasaki and R. Yamashina, "Behavioral cloning from noisy demonstrations," in *International Conference on Learning Representations*, 2020.

[13] O. Nachum and B. Dai, "Reinforcement learning via fenchel-rockafellar duality," *arXiv preprint arXiv:2001.01866*, 2020.

[14] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223.

[15] A. Y. Ng, S. Russell *et al.*, "Algorithms for inverse reinforcement learning." in *Icml*, vol. 1, no. 2, 2000, p. 2.

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

[17] U. Syed and R. E. Schapire, "A game-theoretic approach to apprenticeship learning," *Advances in neural information processing systems*, vol. 20, 2007.

[18] H. Xu, X. Zhan, H. Yin, and H. Qin, "Discriminator-weighted offline imitation learning from suboptimal demonstrations," in *International Conference on Machine Learning*. PMLR, 2022, pp. 24 725–24 742.

[19] S. K. S. Ghasemipour, R. Zemel, and S. Gu, "A divergence minimization perspective on imitation learning methods," in *Conference on robot learning*. PMLR, 2020, pp. 1259–1277.

[20] G.-H. Kim, J. Lee, Y. Jang, H. Yang, and K.-E. Kim, "Lobsdice: Offline learning from observation via stationary distribution correction estimation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 8252–8264, 2022.

[21] P. Dayan, "Improving Generalization for Temporal Difference Learning: The Successor Representation," *Neural Computation*, vol. 5, no. 4, pp. 613–624, Jul. 1993.

[22] A. Barreto, W. Dabney, R. Munos, J. J. Hunt, T. Schaul, H. P. van Hasselt, and D. Silver, "Successor Features for Transfer in Reinforcement Learning," in *Advances in Neural Information Processing Systems*, vol. 30. Curran Associates, Inc., 2017.

[23] M. Yang, S. Levine, and O. Nachum, "Trail: Near-optimal imitation learning with suboptimal data," *arXiv preprint arXiv:2110.14770*, 2021.

[24] A. Ajay, A. Kumar, P. Agrawal, S. Levine, and O. Nachum, "Opal: Offline primitive discovery for accelerating offline reinforcement learning," *arXiv preprint arXiv:2010.13611*, 2020.

[25] S. L. Brunton, M. Budišić, E. Kaiser, and J. N. Kutz, "Modern koopman theory for dynamical systems," *arXiv preprint arXiv:2102.12086*, 2021.

[26] C. Jin, Z. Yang, Z. Wang, and M. I. Jordan, "Provably Efficient Reinforcement Learning with Linear Function Approximation," Aug. 2019, number: arXiv:1907.05388 arXiv:1907.05388 [cs, math, stat].

[27] A. Agarwal, S. Kakade, A. Krishnamurthy, and W. Sun, "FLAMBE: Structural Complexity and Representation Learning of Low Rank MDPs," in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 20 095–20 107.

[28] T. Zhang, T. Ren, M. Yang, J. Gonzalez, D. Schuurmans, and B. Dai, "Making linear mdps practical via contrastive representation learning," in *International Conference on Machine Learning*. PMLR, 2022, pp. 26 447–26 466.

[29] T. Ren, T. Zhang, L. Lee, J. E. Gonzalez, D. Schuurmans, and B. Dai, "Spectral decomposition representation for reinforcement learning," *arXiv preprint arXiv:2208.09515*, 2022.

[30] H. Ma, Z. Ren, B. Dai, and N. Li, "Skill transfer and discovery for sim-to-real learning: A representation-based viewpoint," *arXiv preprint arXiv:2404.05051*, 2024.

[31] M. U. Gutmann and A. Hyvärinen, "Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics." *Journal of machine learning research*, vol. 13, no. 2, 2012.

[32] H. Ma, B. Dai, Z. Ren, Y. Wang, and N. Li, "Offline imitation learning upon sub-optimal demonstrations by primal-dual representation," https://scholar.harvard.edu/haitongma/files/repr-il-online-report.pdf.

[33] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, 2016.

[34] T. Ren, Z. Ren, N. Li, and B. Dai, "Stochastic nonlinear control via finite-dimensional spectral dynamic embedding," in *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE, 2023, pp. 795–800.

[35] A. Devinatz, "Integral representations of positive definite functions," *Transactions of the American Mathematical Society*, vol. 74, no. 1, pp. 56–77, 1953.

[36] 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*. IEEE, 2012, pp. 5026–5033.

[37] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine, "D4rl: Datasets for deep data-driven reinforcement learning," *arXiv preprint arXiv:2004.07219*, 2020.