# Can We Really Learn One Representation to Optimize All Rewards?

**Chongyi Zheng**[*1] **Royina Karegoudra Jayanth**[*1] **Benjamin Eysenbach**[1]
[1]Princeton University
chongyiz@princeton.edu rj5498@princeton.edu

## Abstract

As unsupervised pretraining becomes increasingly ubiquitous in reinforcement learning, a more thorough theoretical understanding of these methods becomes of equal importance to their empirical success. We focus on the setting of unsupervised learning via interaction, where the forward-backward (FB) representation learning serves as a prototypical and popular example. In this paper, we shed light on FB by formally contextualizing the method within a broader class of recent methods that use regression to obtain a low-rank approximation of a successor measure ratio. Our analysis clarifies when FB representations can exist and how the low-rank approximation converges in practice. Building upon the theory, we propose a variant of FB that is both more amenable to theoretical understanding and simpler to optimize in practice. Experiments in didactic settings, as well as in 10 state-based and image-based continuous control domains, demonstrate that our method converges to desired representations with $10^5\times$ smaller errors than FB and improves zero-shot performance by $+24\%$ on average. We also demonstrate that zero-shot policies inferred by our algorithm provide an efficient initialization if the user prefers further fine-tuning on downstream tasks. Our open-source implementation is available in the supplementary materials. Our project website is available at https://chongyi-zheng.github.io/onestep-fb.

## 1 Introduction

Large-scale pre-training has reshaped how we build learning systems in vision [89, 36, 4] and language [100, 1]: a foundation model is trained once on broad data, and then adapted to specific tasks with little or no updates. In the context of RL, such models are known as behavioral foundation models (BFMs) [101, 95, 57], and ideally acquire behavioral knowledge from unsupervised (reward-free) interactions and later specialize to new tasks with minimal additional learning. Similar to large language models (LLMs), this paradigm can be interpreted as in-context learning for RL: the reward in example trajectories induces a prompt, and the pre-trained BFM responds with the optimal behavior directly.

Forward-backward (FB) representation learning [103] is a prominent attempt to realize a BFM. FB proposes to pre-train a pair of representations that can be combined with a downstream reward to obtain a reward-maximizing policy. Prior work based on FB usually contextualizes this method as learning a low-rank approximation of a successor measure ratio [104, 5, 3]. To unpack this promise, we will study when such representations can exist and be learned. Broadly, this paper gets at the question:

*Can one really learn one representation to optimize all rewards in practice?*

In this paper, we extend the theoretical analysis of FB, aiming to provide additional insights into how and why it works in practice. First, we start by examining the assumption that the ground-truth FB

[*]Equal contribution.

43rd International Conference on Machine Learning (ICML 2026). Workshop: RLxF: Reinforcement Learning from World Feedback.

representations always exist, answering the question: *When do the ground-truth FB representations exist?* Answering this question helps us identify the challenge in using low-rank approximation to capture all optimal behaviors. Second, we explicitly reinterpret the FB representation objective as a regression loss, revealing insights into *what the algorithm optimizes* for practitioners. This reinterpretation reveals a connection with fitted Q-evaluation (FQE) [23, 90, 33], motivating us to study the convergence of the low-rank approximation. Third, we therefore construct a new Bellman operator, which is realized by the learning procedure of the FB algorithm. Using this new Bellman operator, we identify another challenge in the convergence of the low-rank approximation. This challenge mainly comes from the circular dependency in FB: the representations and the policy depend on each other.

Building upon our new insights for FB, we propose a simpler alternative to it called **one-step FB**. Our algorithm breaks the circular dependency (Fig. 1) by learning representations for a *fixed* behavioral policy. In doing so, we can pre-train one step of policy improvement over all the behavioral value functions. Through didactic experiments, we demonstrate that FB can struggle to converge, while our proposed variant converges to $10^5$ smaller errors. Experiments on 8 state-based benchmark domains and 2 image-based benchmark domains show that one-step FB is a competitive method for unsupervised pre-training, achieving $+24\%$ improved zero-shot performance on average. In addition, our method provides an efficient initialization for fine-tuning with off-the-shelf RL algorithms. Overall, our method serves as a simpler and more stable plug-and-play alternative to FB, which may appeal to RL practitioners.

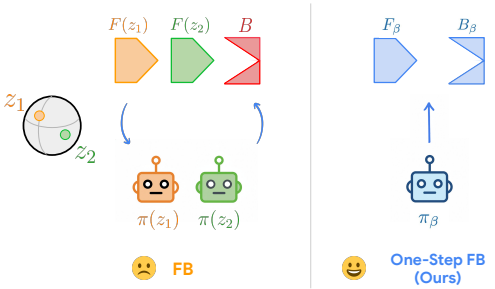



Figure 1: How can we learn a library of policies to quickly maximize new rewards? *(Left)* Forward-backward representation learning (FB) [103] learns bilinear representations to acquire new policies. *(Right)* Our theoretical analysis of this method reveals some optimization challenges, which are alleviated through a simplified method that enjoys stable convergence.

## 2 Preliminary

We first define the notation and background mathematics for our analysis. A conceptual description of the prior related work can be found in Appendix A.

We consider a controlled Markov process (CMP) [10] defined by a state space $\mathcal{S}$, an action space $\mathcal{A}$, a probability measure of initial states $p_0 \in \Delta(\mathcal{S})$, a probability measure of environmental transitions $p : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, and a discount factor $\gamma \in [0, 1)$, where $\Delta(\mathcal{X})$ denotes the set of all possible probability distributions over a space $\mathcal{X}$. When equipped with a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, the CMP becomes a Markov decision process (MDP) [97]. With slight abuse of notation, we use *probability measure* to denote either the probability mass in discrete CMPs or the probability density in continuous CMPs.

**Successor measure and Q-value.** For a CMP and a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, the successor measure [21, 43, 103, 29, 113, 70, 115] $M^\pi : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S} \times \mathcal{A})$ defines the probability measure of reaching a *future* state-action pair $(s_f, a_f)$ starting from a current state-action pair $(s, a)$. Prior work [21, 7, 12] has shown that the successor measure is the unique fixed point of a Bellman equation:

$$M^\pi(s_f, a_f \mid s, a) = (1-\gamma)\delta(s_f, a_f \mid s, a) + \gamma \mathbb{E}_{p(s'|s,a),\, \pi(a'|s')} \left[ M^\pi(s_f, a_f \mid s', a') \right], \quad (1)$$

where $\delta(s_f, a_f \mid s, a)$ is the delta measure[2] centered at the state-action pair $(s, a)$. For a discrete CMP, the successor measure can be written as a matrix $M^\pi \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times|\mathcal{S}\times\mathcal{A}|}$ that is full rank (i.e., $\text{rank}(M^\pi) = |\mathcal{S} \times \mathcal{A}|$) [2, Lemma 1.6 and Corollary 1.5]; see Appendix B.1.

The successor measure can be used to express the Q-value $Q^\pi_r(s, a)$ for any reward function $r$:

$$Q^\pi_r(s, a) = \mathbb{E}_{(s_f,a_f)\sim M^\pi(s_f,a_f|s,a)}[r(s_f, a_f)]. \quad (2)$$

This connection disentangles the estimation of the successor measure (pre-training) and the estimation of the Q-value (fine-tuning) into two separate phases, resembling the learning paradigm in LLMs [16, 78]. Next, we will make this resemblance precise.

[2]The delta measure is an indicator function for discrete MDPs and a Dirac delta function for continuous MDPs.

**Unsupervised pre-training in RL and zero-shot RL.** Algorithms for unsupervised pre-training in RL typically involve two steps: *(Step 1)* pre-training a set of policies and their successor measures in a CMP, and *(Step 2)* performing zero-shot policy adaptation for a specific reward function. The unsupervised pre-training mainly considers the offline setting [53] (Sec. 4), where learning happens on a dataset of transitions $\mathcal{D} = \{(s, a, s', a')\}$ collected by some behavioral policy $\pi_\beta : \mathcal{S} \to \Delta(\mathcal{A})$. We will use zero-shot RL to denote unsupervised pre-training algorithms where policy adaptation does not require updating the neural networks, **independent of acquiring optimal behaviors**. After adapting the zero-shot policy, one can further fine-tune it using off-the-shelf RL algorithms. See Appendix B.2 for additional components in *Step 1* and *Step 2*.

This work aims to analyze and simplify a prior state-of-the-art pre-training method called *forward-backward representation learning* (FB) [103]. We will include an overview of this algorithm next.

**Forward-backward representation learning.** FB is an instance of the zero-shot RL algorithms [5, 81, 115, 3]. During pre-training, FB uses forward-backward representation functions $F(s, a, z)$ and $B(s_f, a_f)$ to parameterize the policy $\pi(a \mid s, z)$ and its successor measure $M^\pi(s_f, a_f \mid s, a, z)$. Importantly, both the latent-conditioned policy and its associated successor measure depend on the forward representation, forming a circular dependency. See Appendix B.3 for the formal definition of ground-truth forward-backward representations $F^\star(s, a, z)$ and $B^\star(s_f, a_f)$. Prior work [103, 104] *assumes* the existence of ground-truth FB representations. This assumption raises the question: *When do the ground-truth FB representations exist?* We will answer this question using tools from linear algebra and rank matching in Sec. 3.1.

After pre-training the latent-conditioned policy and its successor measure, FB finds the optimal policy for any reward function by setting the latent variable to an expectation over backward representations: $z_r = \mathbb{E}_{\rho(s_f, a_f)}[B^\star(s_f, a_f) r(s_f, a_f)]$. See Appendix B.3 for the formal definition. In the following sections, We will also study the convergence of the FB algorithm in practice (Sec. 3.3), motivating us to derive a simpler zero-shot RL algorithm (Sec. 4).

**Least-squares importance fitting.** The goal of probability measure ratio estimation is to predict the ratio $p(x)/q(x)$ between two probability measures $p \in \Delta(\mathcal{X})$ and $q \in \Delta(\mathcal{X})$ over some space $\mathcal{X}$. The most widely adopted approach casts this problem as classification [88, 19, 38]. However, least-squares importance fitting (LSIF) [45, 44] casts the measure ratio estimation as regression.[3] Specifically, LSIF fits a function $g : \mathcal{X} \to \mathbb{R}$ to the target measure ratio $p(x)/q(x)$ using samples.

$$\mathcal{L}_{\text{LSIF}}(g) = \frac{1}{2}\mathbb{E}_{q(x)}\left[\left(g(x) - p(x)/q(x)\right)^2\right] = \frac{1}{2}\mathbb{E}_{q(x)}\left[g(x)^2\right] - \mathbb{E}_{p(x)}\left[g(x)\right] + \text{const.}, \tag{3}$$

where the constant is independent of the learned ratio $g(x)$. Compared with the more popular classification loss, this LSIF loss remains well defined when $g(x)$ is negative. We call $q(x)$ the *anchor* measure and call $p(x)$ the *target* measure. In Sec. 3.2, we will reinterpret the FB representation objective using the LSIF loss function with a special parameterization of the ratio function.

## 3 Understanding Forward-Backward Representation Learning

In this section, we study FB through the lens of linear algebra, LSIF, and contraction mapping, focusing on providing new analysis that extends the theory in FB. The goal of our theoretical analysis is threefold.

- §3.1 We examine the assumption in prior work that ground-truth FB representations always exist, showing strict rank and dimensionality constraints. Our analysis indicates that a low-rank approximation may induce arbitrary errors in capturing optimal behavior for some rewards.
- §3.2 We next reinterpret the FB representation objective as a temporal-difference variant of a regression loss (Eq. 3), drawing a connection with FQE. This connection motivates us to study the convergence of the practical FB.
- §3.3 We construct a new Bellman operator to describe the learning procedure of the practical FB. The failure of applying the Banach fixed-point theorem to this new Bellman operator results in unclear convergence of the practical FB algorithm.

The main challenge in analyzing and providing guarantees for FB lies in the circular dependency (Fig. 1). We thus introduce a variant of FB that breaks the circular (Sec. 4). The resulting algorithm is not only simpler but also enjoys more stable learning and clearer convergence (Sec. 5).

[3] The same loss recurs under different names in the literature [71, 46].

### 3.1 When Do the Ground-Truth FB Representations Exist?

Prior work assumes that the ground-truth FB representations exist (Definition 1 of Touati and Ollivier [103] and Theorem 2 of Touati et al. [104]), raising the question: *When do the ground-truth FB representations exist?* We will explicitly study this question using tools from linear algebra and rank matching (Lemma 1). Specifically, we introduce two sets of necessary conditions for the ground-truth FB representations to hold: Proposition 1 provides insights to understand the practical FB algorithm, and Proposition 2 reveals a criterion for examining its convergence.

The first set of necessary conditions starts with the rank of the ground-truth FB representations. We show that unless the representation dimension is infinitely large, the FB's zero-shot policies are not necessarily optimal for maximizing all rewards.

**Proposition 1** (Informal)**.** *Given any discrete CMP, a finite latent space $\mathcal{Z}$, and a marginal measure $\rho$, any FB representation matrices $F^\star_{\mathcal{Z}} \in \mathbb{R}^{|\mathcal{Z}\times\mathcal{S}\times\mathcal{A}|\times d}$ and $B^\star \in \mathbb{R}^{d\times|\mathcal{S}\times\mathcal{A}|}$ that encode this CMP's successor measure (Definition 2 and Definition 3) must satisfy the following properties:*

1. *The representation dimension $d$ is at least $|\mathcal{S}\times\mathcal{A}|$: $d \geq |\mathcal{S}\times\mathcal{A}|$.*
2. *The rank of the matrix $F^\star_{\mathcal{Z}}$ is at least $|\mathcal{S}\times\mathcal{A}|$ and at most $d$: $|\mathcal{S}\times\mathcal{A}| \leq rank(F^\star_Z) \leq d$.*
3. *The rank of the matrix $B^\star$ equals to $|\mathcal{S}\times\mathcal{A}|$: $rank(B^\star) = |\mathcal{S}\times\mathcal{A}|$.*
4. *The matrix $B^\star$, components of matrix $F^\star_{\mathcal{Z}}$, the successor measures must satisfy:*

$$B^\star = F^{\star+}_{z_1} M^{\pi(a|s,z_1)}/\rho = \cdots = F^{\star+}_{z_{|\mathcal{Z}|}} M^{\pi(a|s,z_{|\mathcal{Z}|})}/\rho,$$

*where $X^+$ denotes the pseudoinverse [69, 11] of the matrix $X$.*

See Appendix C.1 for the complete discussion and a proof. While our rank conditions are derived with finite states and actions, we extend them to continous CMPs and identify an irreducible error in the low-rank approximation used in the practical FB. We discuss two important implications from our Proposition 1 next.

First, our rank analysis implies some challenges in using neural networks to express the FB representations. In machine learning, one usually invokes the Universal Function Approximation Theorem [41] to argue that they can perfectly express a *finite-dimensional* function of interest with large enough neural networks. However, our Proposition 1 suggests that when the states and actions are continuous, both the ground-truth $F^\star$ and $B^\star$ lie in an *infinite-dimensional* Hilbert space [9]. In this case, arbitrarily expressive neural networks still cannot represent the desired representations.

**Corollary 1** (Learnability)**.** *For continuous CMPs with $|\mathcal{S}\times\mathcal{A}| \to \infty$, the inner product $\langle F^\star(s,a,z), B^\star(s,a)\rangle$ lies in an infinite-dimensional Hilbert space $\mathcal{H}$, which contains the latent space $\mathcal{Z} \subset \mathcal{H}$. Thus, neural networks are not able to express the inner product.*

The challenge not only stems from fitting the inner product (one can fit the inner product as a kernel function [9, 61]), but also stems from the infinite-dimensional latent variable $z \in \mathcal{H}$ in the input of the forward representations. As mentioned in prior work [103, 104], using a finite representation dimension results in a low-rank approximation of the ground-truth FB.

Second, when learning a low-rank approximation for the desired ratio (Eq. 12) as in the practical FB, our rank analysis suggests that we can incur arbitrary errors on the optimal Q-value predictions.

**Corollary 2** (Arbitrary Errors; Informal)**.** *For the low-rank FB representations ($d < |\mathcal{S}\times\mathcal{A}|$) learned by the practical FB algorithm, there exist some reward functions such that errors in the optimal Q-value prediction are arbitrarily large.*

See Appendix C.2 for the complete statement and the proof. Unlike the function approximation errors in neural networks [41], these arbitrary errors in the optimal Q-value prediction are irreducible. Therefore, the corresponding zero-shot policy may not be optimal for maximizing the reward.

Our second set of necessary conditions studies the ground-truth forward representations under reward transformations. Recall that the Q-value is equivariant to an arbitrary positive affine transformation on the reward [91, 74] (See Lemma 3). We translate this into an invariance property that the ground-truth forward representations must satisfy:

**Proposition 2** (Informal)**.** *For a scalar $\nu > 0$, and an offset $\xi \in \mathbb{R}$, the ground-truth forward representations $F^\star$ are invariant under affine transformations on the reward, i.e., $F^\star(s,a,z_{\nu r+\xi}) = F^\star(s,a,z_r)$.*

See Appendix C.3 for further discussions and the proof. This proposition underscores a *necessary* condition: any $F$ that failed to satisfy Proposition 2 must not equal $F^\star$ (contrapositive). We will use this proposition as the criterion for examining whether the practical FB converges to the ground-truth representations (Sec. 5.1). In the next section, we reinterpret the representation objective in FB. Our understanding provides insights to simplify the algorithm in Sec. 4.

### 3.2 What Does the FB Representation Objective Minimize?

We now reinterpret the representation objective used in FB. The main idea is to derive a temporal-difference (TD) variant of the LSIF loss (Eq. 3) that minimizes a Bellman error similar to FQE. This TD LSIF loss ends up being equivalent to the representation loss used in Touati and Ollivier [103].

In the context of LSIF, FB chooses to set the ratio function in Eq. 3 as an inner product: $g(s, a, z, s_f, a_f) \triangleq F(s, a, z)^\top B(s_f, a_f)$, set the target measure to $p(s, a, z, s_f, a_f) \triangleq M^\pi(s_f, a_f \mid s, a, z)$, and set the anchor measure to $q(s, a, z, s_f, a_f) \triangleq \rho(s_f, a_f)$, resulting in the following loss:

$$\mathcal{L}_{\text{MC FB}}(F, B) = \frac{1}{2}\mathbb{E}_{\substack{p^{\pi_\beta}(s,a), p_{\mathcal{Z}}(z),\\ \rho(s_f,a_f)}}\left[\left(F(s,a,z)^\top B(s_f,a_f) - \frac{M^\pi(s_f, a_f \mid s, a, z)}{\rho(s_f, a_f)}\right)^2\right]. \quad (4)$$

We call this loss the Monte Carlo (MC) forward-backward representation loss $\mathcal{L}_{\text{MC FB}}$ because, as mentioned in Sec. 2, computing it requires *on-policy* samples from the successor measure $M^\pi$.

We next derive a temporal-difference version of this same loss. First, we replace the successor measure in $\mathcal{L}_{\text{MC FB}}$ using the recursive Bellman equation in Eq. 1. Second, we use target FB representation functions $\bar{F}$ and $\bar{B}$ to replace the ground-truth ratio at the next time step, akin to the target networks used in deep Q-learning [66]. The resulting loss function minimizes a Bellman error[4]:

$$\mathcal{L}_{\text{TD FB}}(F, B) = \frac{1}{2}\mathbb{E}_{\substack{p^{\pi_\beta}(s,a),\, \rho(s_f,a_f),\\ p_{\mathcal{Z}}(z),\, p(s'|s,a),\, \pi(a'|s',z)}}\left[\left(F(s,a,z)^\top B(s_f,a_f) - y\right)^2\right], \quad (5)$$
$$y = (1-\gamma)\delta(s_f, a_f \mid s, a)/\rho(s_f, a_f) + \gamma \bar{F}(s', a', z)^\top \bar{B}(s_f, a_f).$$

See Appendix C.4 for the complete derivation. We call this loss the temporal-difference (TD) forward-backward representation loss $\mathcal{L}_{\text{TD FB}}$. Like FQE, the TD FB loss can be computed using transition samples and target networks. Unlike FQE, this loss function estimates the successor measure ratio instead of the Q-value. Rearranging terms in $\mathcal{L}_{\text{TD FB}}$, we recover the original FB representation objective[5]:

$$\mathcal{L}_{\text{TD FB}}(F, B) = \frac{1}{2}\mathbb{E}_{\substack{p^{\pi_\beta}(s,a),\, \rho(s_f,a_f),\\ p_{\mathcal{Z}}(z),\, p(s'|s,a),\, \pi(a'|s',z)}}\left[\left(F(s,a,z)^\top B(s_f,a_f) - \gamma \bar{F}(s',a',z)^\top \bar{B}(s_f,a_f)\right)^2\right]$$
$$- (1-\gamma)\mathbb{E}_{p^{\pi_\beta}(s,a),\, p_{\mathcal{Z}}(z)}\left[F(s,a,z)^\top B(s,a)\right]. \quad (6)$$

See Appendix C.4 for the derivation. Importantly, we can now interpret the representation objective in FB as performing approximate value iteration, which has a clear connection with the standard Bellman operator and the Banach fixed-point theorem [6]. These theoretical connections motivate us to study whether FB admits a similar convergence guarantee in practice.

### 3.3 Does the Practical FB Algorithm Admit a Stable Convergence?

Our analysis proceeds in two steps. *First*, the resemblance between FB (Eq. 6) and FQE motivates us to define a new Bellman operator, called the FB Bellman operator. Similar to the relationship between the standard Bellman operator and FQE, we interpret the FB algorithm as iteratively applying the FB Bellman operator using samples from the dataset. *Second*, we use the Banach fixed-point theorem [6] to analyze the asymptotic fixed point of the FB Bellman operator, studying whether FB admits approximate convergence.

Similar to Sec. 3.1, we consider discrete CMPs with a finite number of states and actions. We also assume the transition measure $p(s' \mid s, a)$ and the marginal probability measure $\rho(s_f, a_f)$ are known. Under this setup, we can define a new *FB Bellman operator*.

[4] A similar formulation has been mentioned in prior work (See Appendix B of Touati et al. [104]), but from the perspective of minimizing a matrix norm.

[5] We recover the FB representation objective up to a constant scaling factor $1 - \gamma$.

**Definition 1.** *For any two functions $f : \mathcal{S} \times \mathcal{A} \times \mathcal{Z} \to \mathcal{Z}$ and $b : \mathcal{S} \times \mathcal{A} \to \mathcal{Z}$ that induce the latent-conditioned policy $\pi(a \mid s, z) = \delta\left(a \mid \arg\max_{a \in \mathcal{A}} f(s, a, z)^\top z\right)$, the FB Bellman operator $\mathcal{T}_{FB}$ applies to the inner product of $f(s, a, z)$ and $b(s_f, a_f)$:*

$$\mathcal{T}_{FB}\left(f(s,a,z)^\top b(s_f,a_f)\right) \triangleq (1-\gamma)\frac{\delta(s_f, a_f \mid s, a)}{\rho(s_f, a_f)} + \gamma \mathbb{E}_{p(s'|s,a),\pi(a'|s',z)}\left[f(s',a',z)^\top b(s_f,a_f)\right].$$

As discussed in Sec. 3.2, FB's representation objective can be viewed as minimizing a Bellman error (Eq. 5). Comparing the functional form of Eq. 5 to the FB Bellman operator, we can draw a key observation between FB and the FB Bellman operator: the TD FB loss (Eq. 6) in FB is iteratively applying the FB Bellman operator $\mathcal{T}_{\text{FB}}$ to the FB representations from the previous iteration, resembling the bridge between FQE and the standard Bellman operator [66, 58].

The standard Bellman operator is a $\gamma$-contraction and admits a unique fixed point [6]. Unfortunately, the FB Bellman operator is *not* a $\gamma$-contraction because of the circular dependency between the latent-conditioned policy and its associated successor measure (See Definition 2 and Fig. 1).

**Proposition 3** (Informal)**.** *The FB Bellman operator $\mathcal{T}_{FB}$ is not a $\gamma$-contraction. Thus, the Banach fixed-point theorem is not applicable to the FB Bellman operator.*

See Appendix C.5 for the proof. Our analysis does *not* suggest that iteratively applying the FB Bellman operator cannot converge to a fixed point, just that the standard proof strategy is not applicable. Indeed, both our discussion in Sec. 3.1 and prior work [104] have already revealed that there exist *multiple* fixed points for the FB Bellman operator.[6] Therefore, whether the FB algorithm converges stably to a fixed-point remains an open problem. Answering this question might require tools such as the Lefschetz fixed-point theorem [55] or the Lyapunov stability [62], which we leave for future research. One alternative method that might converge is to first fit the changing successor measure using a single network and then conduct bilinear decompositions into FB representations. However, the circular dependency (Fig. 1) persists in this variant. In Sec. 5.1, we will use didactic experiments to demonstrate that FB struggles to converge in practice.

## 4 A Simplified Algorithm for Unsupervised Pre-training in RL

In this section, we derive a variant of FB, building upon our theoretical understanding in Sec. 3. Unlike FB, we take as input a dataset sampled from some *behavioral policy* $\pi_\beta(a \mid s)$ and fit the successor measure ratio of this fixed policy (Fig. 1). Our method is conceptually simpler: pre-training consists of one step of policy improvement over all behavioral value functions on the dataset. Empirically, our proposed variant of FB achieves more stable convergence and higher performance (Sec. 5). Similar to prior work [81, 115, 34], our method also provides an efficient policy initialization for further fine-tuning.

### 4.1 Breaking the Circular Dependency in FB

In the same way that FB uses TD FB loss to fit successor measure ratios, we optimize a forward representation function $F_\beta$ and a backward representation function $B_\beta$ to fit the fixed successor measure ratio of the behavioral policy $\pi_\beta(a \mid s)$. We use notations similar to FB, but $F_\beta$ and $B_\beta$ are semantically different from $F$ and $B$, highlighting the dependency on the behavioral policy using the subscript $\beta$. We also introduce a new TD one-step FB loss to learn the new FB representations:

$$\begin{aligned}\mathcal{L}_{\text{TD one-step FB}}(F_\beta, B_\beta) = \frac{1}{2}\mathbb{E}_{\substack{p^{\pi_\beta}(s,a)\,\rho(s_f,a_f),\\ p(s'|s,a),\,\pi_\beta(a'|s')}}\left[\left(F_\beta(s,a)^\top B_\beta(s_f,a_f) - \gamma \bar{F}_\beta(s',a')^\top \bar{B}_\beta(s_f,a_f)\right)^2\right]\\ - (1-\gamma)\mathbb{E}_{p^{\pi_\beta}(s,a)}\left[F_\beta(s,a)^\top B_\beta(s,a)\right],\end{aligned} \tag{7}$$

where $\bar{F}_\beta$ and $\bar{B}_\beta$ are target representation functions. Unlike the TD FB loss, this TD one-step FB loss samples the next action $a'$ from the behavioral policy $\pi_\beta(a' \mid s')$ and the forward representation function $F_\beta$ does *not* depend on a latent variable. Importantly, the TD one-step FB loss admits a clear fixed-point as the behavioral policy is *fixed*: the learning procedure is a supervised learning problem. In theory, we can also first regress the fixed successor measure ratio and then conduct bilinear decompositions into $F_\beta$ and $B_\beta$, enjoying stable convergence.

[6] For example, applying a rotation (orthonormal) matrix to both FB representations does not change their inner products.

Similar to FB, we additionally regularize the backward representations to be orthonormal [103, 104]:

$$\mathcal{L}_{\text{ortho}}(B_\beta) = \mathbb{E}_{\substack{\rho(s_f,a_f)\\ \rho(s'_f,a'_f)}}\left[(B_\beta(s_f,a_f)^\top B_\beta(s'_f,a'_f))^2 - \|B_\beta(s_f,a_f)\|_2^2 - \left\|B_\beta(s'_f,a'_f)\right\|_2^2\right]. \quad (8)$$

The complete new representation objective contains both the TD one-step FB loss and the orthonormalization regularization, with $\lambda_{\text{ortho}}$ controlling the regularization strength:

$$\mathcal{L}(F_\beta, B_\beta) = \mathcal{L}_{\text{TD one-step FB}}(F_\beta, B_\beta) + \lambda_{\text{ortho}}\mathcal{L}_{\text{ortho}}(B_\beta). \quad (9)$$

In Appendix C.6, we discuss a connection between our method and the singular value decomposition (SVD) of the behavioral successor measure. We will use the learned forward representations to derive a latent-conditioned policy in our algorithm.

### 4.2 Learning a Policy Using Forward Representations

Our approach for learning a policy is similar to FB. Specifically, we select actions to maximize the inner products between the forward representation $F_\beta$ and a latent variable $z$ sampled from the latent prior $p_{\mathcal{Z}}$. For discrete CMPs, we use a softmax policy with temperature $\tau_{\text{policy}}$:

$$\pi(a \mid s, z) = \frac{\exp\left(\tau_{\text{policy}} F_\beta(s,a)^\top z\right)}{\sum_{a' \in \mathcal{A}} \exp\left(\tau_{\text{policy}} F_\beta(s,a')^\top z\right)}, \quad z \sim p_{\mathcal{Z}}(z).$$

For CMPs with continuous actions, we explicitly learn a Gaussian policy by the reparameterized policy gradient trick [39] with an additional behavioral-cloning regularization [31] for the offline setting, with $\lambda_{\text{BC}}$ controlling the regularization strength:

$$\mathcal{L}(\pi) = -\mathbb{E}_{p^{\pi_\beta}(s,a_\beta),\, p_{\mathcal{Z}}(z),\, \pi(a|s,z)}\left[F_\beta(s,a)^\top z + \lambda_{\text{BC}} \log \pi(a_\beta \mid s, z)\right]. \quad (10)$$

After pre-training, we can use the policy and representations to adapt to any reward function, akin to FB. Given a downstream task, we infer the task-specific latent variable $z_r^\beta = \mathbb{E}_{\rho(s_f,a_f)}\left[B_\beta(s_f,a_f) r(s_f,a_f)\right]$ and use it to index the latent-conditioned policy $\pi(a \mid s, z_r^\beta)$. Importantly, this policy adaptation performs one step of policy improvement on the behavioral Q-value (Appendix C.7), similar to the generalized policy improvement in Barreto et al. [7].

**Algorithm summary.** In Alg. 1, we summarize our new algorithm, one-step FB, and our open-source implementation is available online[7]. Starting from the existing FB algorithm, implementing our method makes two simple changes: *(1)* remove the latent variable from the input of the forward representation, and *(2)*, in the representation loss, sample the next action from the dataset instead of the policy. We use neural networks to parameterize the new FB representations $F_\beta^\theta(s,a)$ and $B_\beta^\omega(s_f,a_f)$, and the policy $\pi^\eta(a \mid s, z)$.

## 5 Experiments

Our experiments study whether one-step FB is a simpler and more stable variant of FB. First, we will use a simple discrete CMP to verify the theory in Sec. 3, empirically testing whether FB and one-step FB converge to their desired representations. Second, we compare one-step FB to prior work in standard offline RL and offline-to-online RL benchmarks, measuring zero-shot performance and the benefits of offline fine-tuning. Following prior work [83], all experiments show means and standard deviations across 8 random seeds for state-based tasks (4 random seeds for image-based tasks).

### 5.1 The Failure Mode of the FB Algorithm

*Does the practical FB algorithm converge to the fixed point characterized in Sec. 3.1?* We test the convergence of the FB algorithm by training it for a long time in a simple CMP (Fig. 2) with three states and three actions. We choose this discrete CMP because we can compute the successor measure and the optimal Q-value analytically. Using the MC FB loss (Eq. 4) for FB, which is the analytical analogy of the TD FB loss (Eq. 6), we learn the FB algorithm for $10^5$ gradient steps and then analyze prediction errors and forward KL divergence of the latent-conditioned policy. See Appendix D.1 for implementation details. We will track several metrics (See Appendix D.1 for formal definitions) with the aim of answering the following questions:

[7] https://github.com/chongyi-zheng/onestep-fb

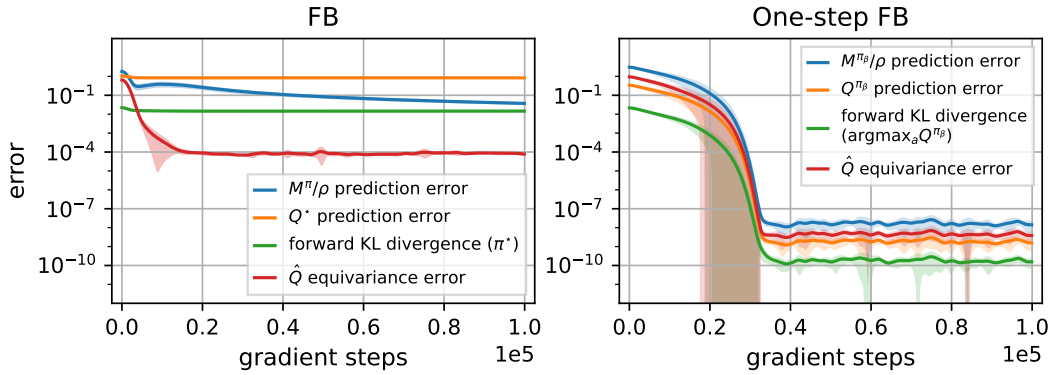



Figure 3: **Learning FB representations in the three-state CMP (Fig. 2).** *(Left)* After training for $10^5$ gradient steps, FB fails to converge to a pair of ground-truth FB representations. *(Right)* Given a fixed policy, one-step FB exactly fits the ground-truth one-step FB representations within $4 \times 10^4$ gradient steps, suggesting that our method is simpler and more stable. These observations are consistent with our theory (Sec. 3.3) and the motivation for developing a new method (Sec. 4.1).

1. Do the learned representations accurately reflect the successor measure ratio? We compute the successor measure ratio prediction error $\epsilon_{\text{SMR}}$.
2. Do the learned representations accurately reflect the Q values of reward-maximizing policies (Definition 3)? We measure this error as the optimal Q-value prediction error $\epsilon_{Q^\star}$.
3. How similar are the learned policies to the reward-maximizing policies (Definition 3)? We measure the forward KL divergence between the latent-conditioned policy and the optimal policy $\text{KL}_{\pi^\star}$.
4. Do the optimal Q-value predictions satisfy the equivariance property of universal value functions (Proposition 2)? We measure the equivariance error of Q predictions $\epsilon_{\text{equiv}}$.

Along several metrics (Fig. 3 *(Left)*), we observe high errors, even asymptotically, suggesting that the FB algorithm might not converge. For example, the prediction error of the successor measure ratio converges to $\epsilon_{\text{SMR}} = 4 \times 10^{-2}$ (contrary to Definition 2). Similarly, the high policy KL divergence ($\text{KL}_{\pi^\star} = 10^{-2}$) indicate that the FB algorithm failed to enable optimal policy adaptation (contrary to Definition 3). Importantly, since the optimal Q-value admits equivariance to an affine transformation, failing to satisfy this property ($\epsilon_{\text{equiv}} = 10^{-4}$) provides key evidence to show that the FB algorithm is *not* converging to the optimal Q-value. In Appendix E.3, we discuss potential confounding effects to clarify the observation that the practical FB algorithm fails to converge.

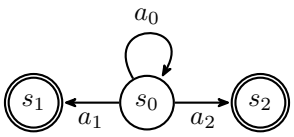



Figure 2: **The three-state CMP.** Agents start from state $s_0$ and take action $a_i$ ($i = 0, 1, 2$) to determinstically transit into state $s_i$. States $s_1$ and $s_2$ are both absorbing states. Sections 5.1 and 5.2 will use this simple MDP to study the convergence of the FB and the one-step FB algorithms.

### 5.2 The Convergence of the One-Step FB Algorithm

We next perform a similar analysis of our simplified algorithm (one-step FB), checking whether the theoretically promised properties (Sec. 4) are borne out in practice. Comparing one-step FB to FB is challenging because they have different fixed points. The closest apples-to-apples comparison is to measure whether one-step FB converges to its respective fixed point (see Sec. 4). For example, we will measure whether the representations learned by one-step FB encode the successor measure of a fixed policy $\pi_\beta(a \mid s)$, not any reward-maximizing policy.

We reuse the three-state CMP in Fig. 2 and train the one-step FB representations for $10^5$ gradient steps using the MC one-step FB loss (Eq. 31). See Appendix D.2 for implementation details. We will track metrics similar to Sec. 5.1, aiming to answer questions related to the Q-value of the fixed policy $Q_r^{\pi_\beta}$ (See Appendix D.2 for formal definitions).

Results in Fig. 3 *(Right)* suggest that all these metrics converge to small numbers ($\leq 10^{-7}$) within $4 \times 10^4$ gradient steps, helping to verify the convergence of one-step FB. In particular, the learned one-step FB representations are equivariant to an affine transformation in rewards ($\epsilon_{\text{equiv}} = 5 \times 10^{-9}$), which is consistent with the property of any Q-value (Lemma 3).

Overall, these didactic experiments show that one-step FB enjoys strong convergence properties. Our next section studies whether these benefits carry over into higher-dimensional continuous control tasks on standard benchmarks.

Table 1: **Zero-shot evaluation on ExORL and OGBench benchmarks.** One-step FB achieves the best or near-best performance on 6 out of 10 domains, outperforming FB by $1.4\times$ on average ($+-$). Following prior work [83], we average results over 8 seeds (4 seeds for image-based tasks) and bold values within 95% of the best performance for each domain. See Table 2 for full results.

| Domain | Laplacian | BYOL-$\gamma$ | ICVF | HILP | FB | One-Step FB |
|---|---|---|---|---|---|---|
| `walker (4 tasks)` | $228 \pm 2$ | $227 \pm 2$ | $\mathbf{619} \pm 23$ | $393 \pm 108$ | $400 \pm 40$ | $379 \pm 26$ (-21) |
| `cheetah (4 tasks)` | $125 \pm 41$ | $127 \pm 39$ | $187 \pm 13$ | $116 \pm 78$ | $271 \pm 46$ | $\mathbf{378} \pm 56$ (+107) |
| `quadruped (4 tasks)` | $462 \pm 35$ | $496 \pm 35$ | $546 \pm 37$ | $352 \pm 59$ | $246 \pm 31$ | $\mathbf{645} \pm 15$ (+399) |
| `jaco (4 tasks)` | $3 \pm 1$ | $3 \pm 0$ | $\mathbf{23} \pm 3$ | $20 \pm 5$ | $10 \pm 4$ | $\mathbf{22} \pm 4$ (+12) |
| `antmaze large navigate (5 tasks)` | $9 \pm 1$ | $21 \pm 2$ | $23 \pm 3$ | $\mathbf{34} \pm 2$ | $25 \pm 5$ | $30 \pm 9$ (+5) |
| `antmaze teleport navigate (5 tasks)` | $3 \pm 1$ | $16 \pm 4$ | $\mathbf{29} \pm 3$ | $19 \pm 6$ | $16 \pm 8$ | $11 \pm 6$ (-5) |
| `cube single play (5 tasks)` | $6 \pm 2$ | $13 \pm 2$ | $13 \pm 2$ | $\mathbf{30} \pm 8$ | $2 \pm 1$ | $3 \pm 2$ (+1) |
| `scene play (5 tasks)` | $4 \pm 1$ | $15 \pm 8$ | $8 \pm 6$ | $\mathbf{19} \pm 6$ | $6 \pm 4$ | $8 \pm 2$ (+2) |
| `visual cube single play (5 tasks)` | - | $11 \pm 4$ | - | $8 \pm 1$ | $12 \pm 3$ | $\mathbf{14} \pm 3$ (+2) |
| `visual scene play (5 tasks)` | - | $3 \pm 1$ | - | $4 \pm 1$ | $13 \pm 2$ | $\mathbf{16} \pm 4$ (+3) |

### 5.3 Comparing One-Step FB to Prior Unsupervised RL Methods

We now compare one-step FB to prior unsupervised RL algorithms, measuring the zero-shot adaptation performance on downstream tasks. While our previous sections have shown that one-step FB enjoys strong convergence properties, one might wonder whether it forgoes some degree of performance by only performing one step of policy improvement. Our experiments will show that, empirically, this is not the case. We defer the rationales for selecting prior methods to Appendix D.3. Appendix D.4 includes the detailed discussions about environments, datasets, and evaluation protocols. Prior work [81, 5, 3] that evaluated on the same benchmarks reported inconsistent results. To make a fair comparison, we implement both our and prior methods from scratch and use the same hyperparameters whenever possible (See Appendix D.5).

We report results in Table 1, aggregating over 4 tasks in each domain of ExORL and 5 tasks in each domain of OGBench, and present the full results in Table 2. These results show that one-step FB matches or surpasses prior unsupervised RL methods on 6 out of 10 domains. In particular, one-step FB achieves $+1.4\times$ improvement over FB on average. On ExORL benchmarks, while prior methods ICVF and HILP are stronger on `walker` domain than both FB and one-step FB, our method performs on par or better than the best-performing baseline in other domains ($+17\%$ on average). On goal-conditioned domains from OGBench, while one-step FB is *not* the best-performing method on state-based tasks, it outperforms prior methods by $20\%$ when taking in pixels as inputs directly. We conjecture that the state-based OGBench domains are challenging for one-step FB because the sparse reward function (goal-conditioned indicator rewards) induces a single backward representation. In contrast, some better-performing baselines are explicitly learning a goal-conditioned distance function, e.g., HILP and ICVF. Taken together, one-step FB is a competitive unsupervised pre-training algorithm for RL. In Appendix E.1, we further study whether one-step FB provides an efficient policy initialization for online fine-tuning.

**Additional experiments.** In Appendix E.2, we study the effects of dataset quality on our algorithm. In Appendix E.3, we investigate the confounding effects in our didactic experiments. Appendix E.4 studies key components of one-step FB: the behavioral-cloning regularization coefficient $\lambda_{\text{BC}}$, the orthonormalization regularization coefficient $\lambda_{\text{ortho}}$, the reward weighting temperature $\tau_{\text{reward}}$, and the representation dimension $d$.

## 6 Conclusion and Limitations

How much computation can the RL algorithm prefetch? The FB framework offers one compelling framework for studying this question, and this paper offers some theoretical considerations on what is required for an unsupervised pre-training method for RL, which led to a simpler method with stronger convergence. Our goal is to help the community interpret, use, and build upon FB-style methods, working towards a future of universal pre-training for RL.

**Limitations.** While we explain why classical fixed-point analysis fails for FB, we do not provide a full alternative convergence theory for the practical FB algorithm. As mentioned at the end of Sec. 3.3, answering this question might require other tools in functional analysis. Practically, the zero-shot policies inferred by one-step FB might be sub-optimal, similar to prior work [81, 115, 34], if obtaining the optimal Q-value requires multiple steps of policy improvement.

## Acknowledgments

We thank Seohong Park for providing helpful feedback on drafts and the project website of this work. This work used the Della computational cluster provided by Princeton Research Computing, as well as the Ionic and Neuronic computing clusters maintained by the Department of Computer Science at Princeton University. This material is based upon work supported by the National Science Foundation under Award No. 2441665. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. Some figures in this work use Twemoji, an open-source emoji set created by Twitter and licensed under CC BY 4.0.

## References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Alekh Agarwal, Nan Jiang, Sham M Kakade, and Wen Sun. Reinforcement learning: Theory and algorithms. *CS Dept., UW Seattle, Seattle, WA, USA, Tech. Rep*, 32:96, 2019.

[3] Siddhant Agarwal, Harshit Sikchi, Peter Stone, and Amy Zhang. Proto successor measure: Representing the behavior space of an RL agent. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=mUDnPzopZF.

[4] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15619–15629, 2023.

[5] Marco Bagatella, Matteo Pirotta, Ahmed Touati, Alessandro Lazaric, and Andrea Tirinzoni. Td-jepa: Latent-predictive representations for zero-shot reinforcement learning. *arXiv preprint arXiv:2510.00739*, 2025.

[6] Stefan Banach. Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales. *Fundamenta mathematicae*, 3(1):133–181, 1922.

[7] André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

[8] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *International conference on machine learning*, pages 531–540. PMLR, 2018.

[9] Alain Berlinet and Christine Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media, 2011.

[10] Abhay G Bhatt and Vivek S Borkar. Occupation measures for controlled markov processes: Characterization and optimality. *The Annals of Probability*, pages 1531–1562, 1996.

[11] Arne Bjerhammar. Application of calculus of matrices to method of least squares: with special reference to geodetic calculations. 1951. URL https://api.semanticscholar.org/CorpusID:118134976.

[12] Léonard Blier, Corentin Tallec, and Yann Ollivier. Learning successor states and goal-dependent values: A mathematical viewpoint. *arXiv preprint arXiv:2101.07123*, 2021.

[13] Diana Borsa, André Barreto, John Quan, Daniel Mankowitz, Hado van Hasselt, Rémi Munos, David Silver, and Tom Schaul. Universal successor features approximators. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=S1VWjiRcKX.

[14] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs. GitHub repository, 2018. URL http://github.com/jax-ml/jax. Version 0.3.13.

[15] David Brandfonbrener, Will Whitney, Rajesh Ranganath, and Joan Bruna. Offline rl without off-policy evaluation. *Advances in neural information processing systems*, 34:4933–4946, 2021.

[16] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[17] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1lJJnR5Ym.

[18] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=a7APmM4B9d.

[19] Kuang Fu Cheng and Chih-Kang Chu. Semiparametric density estimation under a two-sample density ratio model. *Bernoulli*, 10(4):583–604, 2004.

[20] Jongwook Choi, Archit Sharma, Honglak Lee, Sergey Levine, and Shixiang Shane Gu. Variational empowerment as representation learning for goal-conditioned reinforcement learning. In *International conference on machine learning*, pages 1953–1963. PMLR, 2021.

[21] Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural computation*, 5(4):613–624, 1993.

[22] Yiming Ding, Carlos Florensa, Pieter Abbeel, and Mariano Phielipp. Goal-conditioned imitation learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/c8d3a760ebab631565f8509d84b3b3f1-Paper.pdf.

[23] Damien Ernst, Pierre Geurts, and Louis Wehenkel. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6, 2005.

[24] Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International conference on machine learning*, pages 1407–1416. PMLR, 2018.

[25] Ben Eysenbach, Xinyang Geng, Sergey Levine, and Russ R Salakhutdinov. Rewriting history with inverse rl: Hindsight inference for policy improvement. *Advances in neural information processing systems*, 33:14783–14795, 2020.

[26] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. *arXiv preprint arXiv:1802.06070*, 2018.

[27] Benjamin Eysenbach, Matthieu Geist, Ruslan Salakhutdinov, and Sergey Levine. A connection between one-step regularization and critic regularization in reinforcement learning. In *Deep Reinforcement Learning Workshop NeurIPS 2022*, 2022. URL https://openreview.net/forum?id=GXiWE8kDTcn.

[28] Benjamin Eysenbach, Soumith Udatha, Ruslan Salakhutdinov, and Sergey Levine. Imitating past successes can be very suboptimal. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=iqCO3jbPjYF.

[29] Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Russ R Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. *Advances in Neural Information Processing Systems*, 35:35603–35620, 2022.

[30] Kevin Frans, Seohong Park, Pieter Abbeel, and Sergey Levine. Unsupervised zero-shot reinforcement learning via functional reward encodings. In *International Conference on Machine Learning*, pages 13927–13942. PMLR, 2024.

[31] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems*, 34:20132–20145, 2021.

[32] Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pages 1587–1596. PMLR, 2018.

[33] Scott Fujimoto, David Meger, Doina Precup, Ofir Nachum, and Shixiang Shane Gu. Why should i trust you, bellman? the bellman error is a poor replacement for value error. *arXiv preprint arXiv:2201.12417*, 2022.

[34] Dibya Ghosh, Chethan Anand Bhateja, and Sergey Levine. Reinforcement learning from passive data via latent intentions. In *International Conference on Machine Learning*, pages 11321–11339. PMLR, 2023.

[35] Arthur Gretton, Alex Smola, Jiayuan Huang, Marcel Schmittfull, Karsten Borgwardt, Bernhard Schölkopf, et al. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5, 2009.

[36] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[37] Caglar Gulcehre, Ziyu Wang, Alexander Novikov, Tom Le Paine, Sergio Gómez Colmenarejo, Konrad Zolna, Rishabh Agarwal, Josh Merel, Daniel Mankowitz, Cosmin Paduraru, Gabriel Dulac-Arnold, Jerry Li, Mohammad Norouzi, Matt Hoffman, Ofir Nachum, George Tucker, Nicolas Heess, and Nando deFreitas. Rl unplugged: Benchmarks for offline reinforcement learning, 2020.

[38] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

[39] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. PMLR, 2018.

[40] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2023. URL https://arxiv.org/abs/1606.08415.

[41] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[42] Hao Hu, Yiqin Yang, Jianing Ye, Ziqing Mai, and Chongjie Zhang. Unsupervised behavior extraction via random intent priors. *Advances in Neural Information Processing Systems*, 36: 51491–51514, 2023.

[43] Michael Janner, Igor Mordatch, and Sergey Levine. Gamma-models: Generative temporal difference learning for infinite-horizon prediction. *Advances in neural information processing systems*, 33:1724–1735, 2020.

[44] Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. Efficient direct density ratio estimation for non-stationarity adaptation and outlier detection. *Advances in neural information processing systems*, 21, 2008.

[45] Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. A least-squares approach to direct importance estimation. *The Journal of Machine Learning Research*, 10:1391–1445, 2009.

[46] Masahiro Kato. Riesz regression as direct density ratio estimation. *arXiv preprint arXiv:2511.04568*, 2025.

[47] Junsu Kim, Seohong Park, and Sergey Levine. Unsupervised-to-online reinforcement learning. *arXiv preprint arXiv:2408.14785*, 2024.

[48] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[49] Alexander S Klyubin, Daniel Polani, and Chrystopher L Nehaniv. Empowerment: A universal agent-centric measure of control. In *2005 ieee congress on evolutionary computation*, volume 1, pages 128–135. IEEE, 2005.

[50] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

[51] Tejas D Kulkarni, Ardavan Saeedi, Simanta Gautam, and Samuel J Gershman. Deep successor reinforcement learning. *arXiv preprint arXiv:1606.02396*, 2016.

[52] Aviral Kumar, Xue Bin Peng, and Sergey Levine. Reward-conditioned policies, 2019. URL https://arxiv.org/abs/1912.13465.

[53] Sascha Lange, Thomas Gabel, and Martin Riedmiller. Batch reinforcement learning. In *Reinforcement learning: State-of-the-art*, pages 45–73. Springer, 2012.

[54] Daniel Lawson, Adriana Hugessen, Charlotte Cloutier, Glen Berseth, and Khimya Khetarpal. Self-predictive representations for combinatorial generalization in behavioral cloning, 2025. URL https://arxiv.org/abs/2506.10137.

[55] Solomon Lefschetz. Intersections and transformations of complexes and manifolds. *Transactions of the American Mathematical Society*, 28(1):1–49, 1926.

[56] Alexander Li, Lerrel Pinto, and Pieter Abbeel. Generalized hindsight for reinforcement learning. *Advances in Neural Information Processing Systems*, 33, 2020.

[57] Yitang Li, Zhengyi Luo, Tonghe Zhang, Cunxi Dai, Anssi Kanervisto, Andrea Tirinzoni, Haoyang Weng, Kris Kitani, Mateusz Guzek, Ahmed Touati, et al. Bfm-zero: A promptable behavioral foundation model for humanoid control using unsupervised reinforcement learning. *arXiv preprint arXiv:2511.04131*, 2025.

[58] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[59] David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. *arXiv preprint arXiv:1610.06545*, 2016.

[60] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[61] Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*, 2019.

[62] Aleksandr Mikhailovich Lyapunov. The general problem of the stability of motion. *International journal of control*, 55(3):531–534, 1992.

[63] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. VIP: Towards universal visual reward and representation via value-implicit pre-training. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=YJ7o2wetJ2.

[64] Zhuang Ma and Michael Collins. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. *arXiv preprint arXiv:1809.01812*, 2018.

[65] Bogdan Mazoure, Benjamin Eysenbach, Ofir Nachum, and Jonathan Tompson. Contrastive value learning: Implicit models for simple offline RL. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=oqOfLP6bJy.

[66] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[67] Shakir Mohamed and Danilo Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. *Advances in neural information processing systems*, 28, 2015.

[68] I Momennejad, EM Russek, JH Cheong, MM Botvinick, ND Daw, and SJ Gershman. The successor representation in human reinforcement learning. *Nature Human Behaviour*, 1(9): 680–692, 2017. doi: 10.1038/s41562-017-0180-8.

[69] Eliakim H Moore. On the reciprocal of the general algebraic matrix. *Bulletin of the american mathematical society*, 26:294–295, 1920.

[70] Vivek Myers, Chongyi Zheng, Anca Dragan, Sergey Levine, and Benjamin Eysenbach. Learning temporal distances: Contrastive successor features can provide a metric structure for decision-making. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=xQiYCmDrjp.

[71] Ofir Nachum, Yinlam Chow, Bo Dai, and Lihong Li. Dualdice: Behavior-agnostic estimation of discounted stationary distribution corrections. *Advances in neural information processing systems*, 32, 2019.

[72] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 807–814, 2010.

[73] Mitsuhiko Nakamoto, Yuexiang Zhai, Anikait Singh, Max Sobol Mark, Yi Ma, Chelsea Finn, Aviral Kumar, and Sergey Levine. Cal-QL: Calibrated offline RL pre-training for efficient online fine-tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=GcEIvidYSw.

[74] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pages 278–287. Citeseer, 1999.

[75] XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

[76] Tianwei Ni, Benjamin Eysenbach, Erfan Seyedsalehi, Michel Ma, Clement Gehring, Aditya Mahajan, and Pierre-Luc Bacon. Bridging state and history representations: Understanding self-predictive rl. *arXiv preprint arXiv:2401.08898*, 2024.

[77] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[78] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[79] Seohong Park, Oleh Rybkin, and Sergey Levine. Metra: Scalable unsupervised rl with metric-aware abstraction. *arXiv preprint arXiv:2310.08887*, 2023.

[80] Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. Ogbench: Benchmarking offline goal-conditioned rl. *arXiv preprint arXiv:2410.20092*, 2024.

[81] Seohong Park, Tobias Kreiman, and Sergey Levine. Foundation policies with hilbert representations. In *International Conference on Machine Learning*, pages 39737–39761. PMLR, 2024.

[82] Seohong Park, Kevin Frans, Deepinder Mann, Benjamin Eysenbach, Aviral Kumar, and Sergey Levine. Horizon reduction makes rl scalable. *arXiv preprint arXiv:2506.04168*, 2025.

[83] Seohong Park, Qiyang Li, and Sergey Levine. Flow q-learning. *arXiv preprint arXiv:2502.02538*, 2025.

[84] Seohong Park, Aditya Oberai, Pranav Atreya, and Sergey Levine. Transitive rl: Value learning via divide and conquer. *arXiv preprint arXiv:2510.22512*, 2025.

[85] Jan Peters and Stefan Schaal. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th international conference on Machine learning*, pages 745–750, 2007.

[86] Jan Peters, Katharina Mülling, and Yasemin Altün. Relative entropy policy search. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, AAAI'10, page 1607–1612. AAAI Press, 2010.

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

[88] Jing Qin. Inferences for case-control and semiparametric two-sample density ratio models. *Biometrika*, 85(3):619–630, 1998.

[89] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[90] Martin Riedmiller. Neural fitted q iteration–first experiences with a data efficient neural reinforcement learning method. In *European conference on machine learning*, pages 317–328. Springer, 2005.

[91] Stuart Russell, Peter Norvig, and Artificial Intelligence. A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25(27):79–80, 1995.

[92] Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation. *arXiv preprint arXiv:1803.00653*, 2018.

[93] Tom Schaul, Daniel Horgan, Karol Gregor, and David Silver. Universal value function approximators. In *International conference on machine learning*, pages 1312–1320. PMLR, 2015.

[94] Devan Shah, Owen Yang, Daniel Yang, Chongyi Zheng, and Benjamin Eysenbach. Structured response diversity with mutual information. In *Workshop on Scaling Environments for Agents*, 2025. URL https://openreview.net/forum?id=xL7Bt4jS2U.

[95] Harshit Sikchi, Andrea Tirinzoni, Ahmed Touati, Yingchen Xu, Anssi Kanervisto, Scott Niekum, Amy Zhang, Alessandro Lazaric, and Matteo Pirotta. Fast adaptation with behavioral foundation models. In *Reinforcement Learning Conference*, 2025. URL https://openreview.net/forum?id=soeW8RGo1N.

[96] Masashi Sugiyama, Taiji Suzuki, Shinichi Nakajima, Hisashi Kashima, Paul Von Bünau, and Motoaki Kawanabe. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746, 2008.

[97] Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

[98] Denis Tarasov, Alexander Nikulin, Dmitry Akimov, Vladislav Kurenkov, and Sergey Kolesnikov. Corl: Research-oriented deep offline reinforcement learning library. *Advances in Neural Information Processing Systems*, 36:30997–31020, 2023.

[99] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

[100] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[101] Andrea Tirinzoni, Ahmed Touati, Jesse Farebrother, Mateusz Guzek, Anssi Kanervisto, Yingchen Xu, Alessandro Lazaric, and Matteo Pirotta. Zero-shot whole-body humanoid control via behavioral foundation models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=9sOR0nYLtz.

[102] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[103] Ahmed Touati and Yann Ollivier. Learning one representation to optimize all rewards. *Advances in Neural Information Processing Systems*, 34:13–23, 2021.

[104] Ahmed Touati, Jérémy Rapin, and Yann Ollivier. Does zero-shot reinforcement learning exist? In *The Eleventh International Conference on Learning Representations*, 2022.

[105] Qing Wang, Jiechao Xiong, Lei Han, Han Liu, Tong Zhang, et al. Exponentially weighted imitation learning for batched historical data. *Advances in Neural Information Processing Systems*, 31, 2018.

[106] Kathryn Wantlin, Chongyi Zheng, and Benjamin Eysenbach. Consistent zero-shot imitation with contrastive goal inference. *arXiv preprint arXiv:2510.17059*, 2025.

[107] Yifan Wu, George Tucker, and Ofir Nachum. The laplacian in rl: Learning representations with efficient approximations. *arXiv preprint arXiv:1810.04586*, 2018.

[108] Eva Yi Xie, Nathaniel D. Daw, and Benjamin Eysenbach. Low-rank successor representations capture human-like generalization. In *UniReps: 3rd Edition of the Workshop on Unifying Representations in Neural Models*, 2025. URL https://openreview.net/forum?id=nJJtcmDVvp.

[109] Denis Yarats, David Brandfonbrener, Hao Liu, Michael Laskin, Pieter Abbeel, Alessandro Lazaric, and Lerrel Pinto. Don't change the algorithm, change the data: Exploratory data for offline reinforcement learning. *arXiv preprint arXiv:2201.13425*, 2022.

[110] Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=-2FCwDKRREu.

[111] Jingwei Zhang, Jost Tobias Springenberg, Joschka Boedecker, and Wolfram Burgard. Deep reinforcement learning with successor features for navigation across similar environments. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2371–2378. IEEE, 2017.

[112] Chongyi Zheng, Benjamin Eysenbach, Homer Walke, Patrick Yin, Kuan Fang, Ruslan Salakhutdinov, and Sergey Levine. Stabilizing contrastive rl: Techniques for robotic goal reaching from offline data. *arXiv preprint arXiv:2306.03346*, 2023.

[113] Chongyi Zheng, Ruslan Salakhutdinov, and Benjamin Eysenbach. Contrastive difference predictive coding. *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=0akLDTFR9x.

[114] Chongyi Zheng, Jens Tuyls, Joanne Peng, and Benjamin Eysenbach. Can a misl fly? analysis and ingredients for mutual information skill learning. *arXiv preprint arXiv:2412.08021*, 2024.

[115] Chongyi Zheng, Seohong Park, Sergey Levine, and Benjamin Eysenbach. Intention-conditioned flow occupancy models. *arXiv preprint arXiv:2506.08902*, 2025.

[116] Zhiyuan Zhou, Andy Peng, Qiyang Li, Sergey Levine, and Aviral Kumar. Efficient online reinforcement learning fine-tuning need not retain offline data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=HNOCYZbAPw.

**Algorithm 1** One-step forward-backward representation learning

**input**: The dataset $\mathcal{D}$, the forward representation $F_\beta^\theta$, the backward representation $B_\beta^\omega$, the latent-conditioned policy $\pi^\eta$, the latent prior $p_\mathcal{Z}$, the target forward representation $F_\beta^{\bar{\theta}}$, the target backward representation $B_\beta^{\bar{\omega}}$.
**for** each iteration **do**
  Sample a batch of transitions $\{(s, a, s', a')\} \sim \mathcal{D}$ and a batch of latents $\{z\} \sim p_\mathcal{Z}(z)$.
  Train the forward representation $F_\beta^\theta$ and the backward representation $B_\beta^\omega$ by minimizing $\mathcal{L}(\theta, \omega)$ (Eq. 9).
  Train the policy $\pi^\eta$ by minimizing $\mathcal{L}(\eta)$ (Eq. 10).
  Update the target forward representations $F_\beta^{\bar{\theta}}$ and the target backward representations $B_\beta^{\bar{\omega}}$ using Polyak averages.
**end for**
**output**: $F_\beta^\theta$, $B_\beta^\omega$, and $\pi^\eta$.

Table 2: **Zero-shot evaluation on ExORL and OGbench benchmarks.** We present the full zero-shot evaluation results on 16 ExORL tasks and 30 OGbench tasks. In each domain, we pre-train different methods and evaluate zero-shot performance on a set of tasks. We aggregate the results over 8 seeds (4 seeds for image-based tasks) and bold values within 95% of the best performance for each task.

| Domain | Task | Laplacian | BYOL-$\gamma$ | ICVF | HILP | FB | One-Step FB |
|---|---|---|---|---|---|---|---|
| walker | overall | $228 \pm 2$ | $227 \pm 2$ | $\mathbf{619} \pm 23$ | $393 \pm 108$ | $400 \pm 40$ | $379 \pm 26$ (-21) |
| | flip | $243 \pm 5$ | $242 \pm 4$ | $\mathbf{538} \pm 19$ | $332 \pm 135$ | $277 \pm 100$ | $388 \pm 30$ |
| | run | $89 \pm 1$ | $89 \pm 2$ | $\mathbf{258} \pm 20$ | $136 \pm 36$ | $194 \pm 20$ | $198 \pm 30$ |
| | stand | $389 \pm 3$ | $387 \pm 5$ | $\mathbf{858} \pm 31$ | $691 \pm 126$ | $621 \pm 93$ | $524 \pm 60$ |
| | walk | $192 \pm 4$ | $192 \pm 4$ | $\mathbf{821} \pm 40$ | $413 \pm 195$ | $506 \pm 34$ | $407 \pm 63$ |
| cheetah | overall | $125 \pm 41$ | $127 \pm 39$ | $187 \pm 13$ | $116 \pm 78$ | $271 \pm 46$ | $\mathbf{378} \pm 56$ (+107) |
| | run | $40 \pm 13$ | $42 \pm 13$ | $89 \pm 7$ | $38 \pm 32$ | $43 \pm 36$ | $\mathbf{97} \pm 19$ |
| | run backward | $50 \pm 17$ | $48 \pm 15$ | $48 \pm 9$ | $36 \pm 40$ | $125 \pm 30$ | $\mathbf{181} \pm 60$ |
| | walk | $185 \pm 61$ | $199 \pm 61$ | $385 \pm 21$ | $195 \pm 121$ | $251 \pm 166$ | $\mathbf{424} \pm 44$ |
| | walk backward | $226 \pm 77$ | $220 \pm 70$ | $226 \pm 38$ | $194 \pm 202$ | $663 \pm 143$ | $\mathbf{811} \pm 156$ |
| quadruped | overall | $462 \pm 35$ | $496 \pm 35$ | $546 \pm 37$ | $352 \pm 59$ | $246 \pm 31$ | $\mathbf{645} \pm 15$ (+399) |
| | jump | $554 \pm 54$ | $603 \pm 67$ | $617 \pm 59$ | $321 \pm 63$ | $247 \pm 84$ | $\mathbf{707} \pm 48$ |
| | run | $324 \pm 22$ | $345 \pm 19$ | $395 \pm 33$ | $277 \pm 56$ | $165 \pm 51$ | $\mathbf{461} \pm 11$ |
| | stand | $651 \pm 47$ | $700 \pm 41$ | $796 \pm 57$ | $473 \pm 103$ | $388 \pm 86$ | $\mathbf{916} \pm 30$ |
| | walk | $318 \pm 21$ | $337 \pm 13$ | $375 \pm 57$ | $339 \pm 111$ | $183 \pm 59$ | $\mathbf{496} \pm 27$ |
| jaco | overall | $3 \pm 1$ | $3 \pm 0$ | $\mathbf{23} \pm 3$ | $20 \pm 5$ | $10 \pm 4$ | $\mathbf{22} \pm 4$ (+12) |
| | reach bottom left | $3 \pm 1$ | $3 \pm 1$ | $25 \pm 9$ | $\mathbf{29} \pm 9$ | $8 \pm 5$ | $6 \pm 2$ |
| | reach bottom right | $3 \pm 0$ | $3 \pm 1$ | $21 \pm 8$ | $\mathbf{24} \pm 8$ | $7 \pm 9$ | $5 \pm 2$ |
| | reach top left | $2 \pm 1$ | $3 \pm 0$ | $26 \pm 10$ | $6 \pm 8$ | $9 \pm 9$ | $\mathbf{53} \pm 11$ |
| | reach top right | $4 \pm 1$ | $3 \pm 1$ | $20 \pm 7$ | $\mathbf{22} \pm 11$ | $17 \pm 9$ | $\mathbf{22} \pm 6$ |
| antmaze large navigate | overall | $9 \pm 1$ | $21 \pm 2$ | $23 \pm 3$ | $\mathbf{34} \pm 2$ | $25 \pm 5$ | $30 \pm 9$ (+5) |
| | task 1 | $2 \pm 1$ | $6 \pm 3$ | $4 \pm 2$ | $13 \pm 8$ | $\mathbf{46} \pm 9$ | $21 \pm 9$ |
| | task 2 | $2 \pm 1$ | $11 \pm 4$ | $9 \pm 3$ | $16 \pm 6$ | $2 \pm 3$ | $\mathbf{41} \pm 12$ |
| | task 3 | $29 \pm 3$ | $57 \pm 8$ | $67 \pm 8$ | $\mathbf{75} \pm 6$ | $31 \pm 10$ | $15 \pm 4$ |
| | task 4 | $6 \pm 2$ | $14 \pm 5$ | $18 \pm 6$ | $27 \pm 10$ | $3 \pm 2$ | $\mathbf{33} \pm 15$ |
| | task 5 | $6 \pm 3$ | $16 \pm 4$ | $18 \pm 4$ | $40 \pm 8$ | $\mathbf{44} \pm 19$ | $37 \pm 20$ |
| antmaze teleport navigate | overall | $3 \pm 1$ | $16 \pm 4$ | $\mathbf{29} \pm 3$ | $19 \pm 6$ | $16 \pm 8$ | $11 \pm 6$ (-5) |
| | task 1 | $1 \pm 1$ | $5 \pm 3$ | $\mathbf{17} \pm 9$ | $10 \pm 5$ | $8 \pm 9$ | $2 \pm 1$ |
| | task 2 | $6 \pm 2$ | $15 \pm 7$ | $\mathbf{38} \pm 17$ | $27 \pm 7$ | $11 \pm 10$ | $15 \pm 10$ |
| | task 3 | $4 \pm 2$ | $15 \pm 3$ | $\mathbf{40} \pm 4$ | $24 \pm 8$ | $23 \pm 10$ | $17 \pm 10$ |
| | task 4 | $2 \pm 1$ | $24 \pm 12$ | $\mathbf{40} \pm 6$ | $21 \pm 11$ | $25 \pm 10$ | $19 \pm 11$ |
| | task 5 | $2 \pm 1$ | $\mathbf{23} \pm 5$ | $12 \pm 11$ | $14 \pm 7$ | $13 \pm 12$ | $2 \pm 1$ |
| cube single play | overall | $6 \pm 2$ | $13 \pm 2$ | $13 \pm 2$ | $\mathbf{30} \pm 8$ | $2 \pm 1$ | $3 \pm 2$ (+1) |
| | task 1 | $5 \pm 2$ | $12 \pm 4$ | $13 \pm 2$ | $\mathbf{27} \pm 7$ | $1 \pm 1$ | $3 \pm 4$ |
| | task 2 | $5 \pm 2$ | $13 \pm 3$ | $13 \pm 3$ | $\mathbf{30} \pm 10$ | $3 \pm 2$ | $4 \pm 4$ |
| | task 3 | $7 \pm 3$ | $13 \pm 4$ | $14 \pm 4$ | $\mathbf{23} \pm 13$ | $3 \pm 3$ | $4 \pm 4$ |
| | task 4 | $6 \pm 2$ | $15 \pm 6$ | $11 \pm 3$ | $\mathbf{37} \pm 19$ | $2 \pm 2$ | $2 \pm 2$ |
| | task 5 | $5 \pm 3$ | $11 \pm 3$ | $13 \pm 4$ | $\mathbf{30} \pm 22$ | $1 \pm 2$ | $1 \pm 1$ |
| scene play | overall | $4 \pm 1$ | $15 \pm 8$ | $8 \pm 6$ | $\mathbf{19} \pm 6$ | $6 \pm 4$ | $8 \pm 2$ (+2) |
| | task 1 | $17 \pm 6$ | $49 \pm 32$ | $34 \pm 23$ | $\mathbf{66} \pm 16$ | $25 \pm 16$ | $21 \pm 8$ |
| | task 2 | $1 \pm 1$ | $11 \pm 8$ | $5 \pm 7$ | $\mathbf{14} \pm 11$ | $5 \pm 5$ | $12 \pm 4$ |
| | task 3 | $1 \pm 1$ | $9 \pm 6$ | $3 \pm 3$ | $\mathbf{12} \pm 14$ | $0 \pm 1$ | $0 \pm 0$ |
| | task 4 | $2 \pm 1$ | $4 \pm 7$ | $0 \pm 0$ | $1 \pm 1$ | $0 \pm 0$ | $\mathbf{7} \pm 4$ |
| | task 5 | $\mathbf{0} \pm \mathbf{0}$ | $\mathbf{0} \pm \mathbf{0}$ | $\mathbf{0} \pm \mathbf{0}$ | $\mathbf{0} \pm \mathbf{0}$ | $\mathbf{0} \pm \mathbf{0}$ | $\mathbf{0} \pm \mathbf{0}$ |
| visual cube single play | overall | - | $11 \pm 4$ | - | $8 \pm 1$ | $12 \pm 3$ | $\mathbf{14} \pm 3$ (+2) |
| | task 1 | - | $24 \pm 16$ | - | $10 \pm 6$ | $14 \pm 7$ | $\mathbf{47} \pm 12$ |
| | task 2 | - | $10 \pm 2$ | - | $\mathbf{19} \pm 5$ | $10 \pm 5$ | $11 \pm 3$ |
| | task 3 | - | $\mathbf{16} \pm 4$ | - | $10 \pm 7$ | $\mathbf{15} \pm 6$ | $3 \pm 3$ |
| | task 4 | - | $3 \pm 3$ | - | $\mathbf{13} \pm 7$ | $8 \pm 5$ | $9 \pm 8$ |
| | task 5 | - | $1 \pm 1$ | - | $\mathbf{15} \pm 3$ | $11 \pm 7$ | $2 \pm 2$ |
| visual scene play | overall | - | $11 \pm 4$ | - | $8 \pm 1$ | $12 \pm 3$ | $\mathbf{14} \pm 3$ (+2) |
| | task 1 | - | $7 \pm 2$ | - | $14 \pm 4$ | $66 \pm 12$ | $\mathbf{78} \pm 16$ |
| | task 2 | - | $1 \pm 1$ | - | $2 \pm 1$ | $0 \pm 0$ | $\mathbf{3} \pm 4$ |
| | task 3 | - | $0 \pm 1$ | - | $\mathbf{2} \pm 1$ | $0 \pm 0$ | $0 \pm 0$ |
| | task 4 | - | $\mathbf{5} \pm 5$ | - | $0 \pm 0$ | $0 \pm 0$ | $1 \pm 1$ |
| | task 5 | - | $\mathbf{0} \pm 1$ | - | $\mathbf{0} \pm 1$ | $\mathbf{0} \pm 0$ | $\mathbf{0} \pm 0$ |

# A Related Work

Our work investigates the theoretical foundations of unsupervised pre-training in RL, with a focus on the prior forward-backward (FB) representation learning algorithm [103].

**Unsupervised RL and zero-shot RL.** The broader goal of unsupervised RL is to pre-train policies from reward-free *unsupervised interactions* that enable efficient adaptation to downstream tasks. Prior work has approached this via skill learning [30, 81, 47, 42, 79, 26, 114], intent predictions [30, 115], empowerment maximization [49, 20, 67, 94], or self-supervised representation learning [76, 63, 65, 110, 112]. After pre-training a set of policies, these methods typically adapt one of the policies to a new reward function via continuous fine-tuning or hierarchical control [79, 30, 63, 115]. One appealing family of methods that does not require gradient-based fine-tuning of the pre-trained policy is called zero-shot RL methods [103, 81, 5, 101, 57]. Similar to the in-context learning in LLMs [16], zero-shot RL methods prefetch optimal policies for *any* rewards during pre-training and perform in-context adaptation on downstream tasks. In this paper, through theoretical and empirical analysis, we demystify a prior SOTA zero-shot RL method (FB [103]) and study its convergence in practice.

**Successor measures and successor features.** Our work builds on successor measures [21], which were originally proposed to improve generalization in RL and have since been widely adopted in neuroscience to model predictive maps in the brain [68, 108]. In the domain of Deep RL, prior work has shown that successor measures can be learned in high-dimensional environments [51, 111] and facilitate transfer learning across tasks [7]. By combining these ideas with universal value function approximators [93], Universal Successor Features (USFs) generalize successor features to estimate values for any reward under any policy [13]. More recently, forward-backward (FB) representation learning [103] extended this to enable zero-shot evaluation for *any* reward function, forming the basis for building behavioral foundation models [101, 5]. Our analysis of FB interprets the representation objective as estimating the successor measure of a latent-conditioned policy. However, this estimation incurs a circular dependency.

**Density ratio estimations.** Directly estimating the ratios between two probability density functions is an important problem in machine learning. Solving this problem enables applications in two-sample testing [59], covariate shift adaptation [96], outlier detection [96], mutual information estimation [8, 87], and policy evaluation [71]. Prior work has tackled the density ratio estimation problem by minimizing a KL divergence [96], moment matching [35], penalized convex risk minimization [75], contrastive learning [64, 77, 87]. Our analysis of the FB algorithm is closely related to the least-squares importance fitting approach [44, 45] for density ratio estimation. In this paper, we show that the FB representation objective is a temporal-difference variant of the least-squares importance fitting loss in Sec. 3.2 and is closely related to fitted Q-evaluation (FQE) [90].

**One-step RL.** One-step RL methods [15, 37, 85, 105, 86] apply one step of policy improvement to a data-generating behavioral policy. These methods have two phases: First, estimate the Q-values of the behavioral policy via regression or FQE updates. Second, optimize the policy to maximize the predicted Q-value. This formulation decouples Q-value estimation from policy extraction, encompassing a wide range of techniques, from Relative Entropy Policy Search [86] to goal-conditioned imitation learning [92, 22, 56, 18, 52, 25, 28, 106]. Theoretical and empirical analysis presented in Eysenbach et al. [27] show that one step of policy improvement is equivalent to multi-step critic regularization, opening the door to developing a simpler suite of algorithms. Recent work [29, 84, 15] has applied the idea of one-step policy improvement to develop practical RL algorithms. Similarly, our proposed method, one-step FB, adopts the principle of one-step policy improvement, breaking the circular dependency in the original FB algorithm.

# B Preliminary

## B.1 The Successor Measure Matrix

For policy $\pi$, we define the policy-dependent transition matrix $P^\pi \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times|\mathcal{S}\times\mathcal{A}|}$ as $P^\pi_{(s,a),(s',a')} = p(s' \mid s,a)\pi(a' \mid s')$.

**Lemma 1** (Lemma 1.6 and Corollary 1.5 of Agarwal et al. [2]). *For any policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, transition function $p(s' \mid s, a)$ and discount $\gamma \in [0, 1)$, the successor measure matrix $M^\pi \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times|\mathcal{S}\times\mathcal{A}|}$ satisfies $M^\pi = (1-\gamma)(I_{|\mathcal{S}\times\mathcal{A}|} - \gamma P^\pi)^{-1}$. Furthermore, $M^\pi$ is full rank, i.e., rank$(M^\pi) = |\mathcal{S} \times \mathcal{A}|$.*

*Proof.* This result is almost an immediate consequence of the Bellman equation in Eq. 1:

$$M^\pi = (1-\gamma)I_{|\mathcal{S}\times\mathcal{A}|} + \gamma P^\pi M^\pi \implies (I_{|\mathcal{S}\times\mathcal{A}|} - \gamma P^\pi)M^\pi = (1-\gamma)I_{|\mathcal{S}\times\mathcal{A}|}.$$

If $I_{|\mathcal{S}\times\mathcal{A}|} - \gamma P^\pi$ is invertible (i.e., full rank), then successor measure must satisfy $M^\pi = (1-\gamma)(I_{|\mathcal{S}\times\mathcal{A}|} - \gamma P^\pi)^{-1}$. Thus, the proof boils down to showing that this matrix is invertible.

We prove that the matrix $I_{|\mathcal{S}\times\mathcal{A}|} - \gamma P^\pi$ is invertible by showing its null space only contains the zero vector. For any non-zero vector $x \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$, the $L^\infty$-norm of $(I_{|\mathcal{S}\times\mathcal{A}|} - \gamma P^\pi)x$ satisfies

$$\begin{aligned}
\|(I_{|\mathcal{S}\times\mathcal{A}|} - \gamma P^\pi)x\|_\infty &= \|x - \gamma P^\pi x\|_\infty \\
&\overset{(a)}{\geq} \|x\|_\infty - \gamma\|P^\pi x\|_\infty \\
&\overset{(b)}{\geq} \|x\|_\infty - \gamma\|x\|_\infty \\
&= (1-\gamma)\|x\|_\infty \\
&\overset{(c)}{>} 0,
\end{aligned}$$

where, in *(a)*, we apply the triangle inequality of the $L^\infty$-norm, *(b)* holds because $P^\pi x$ is an expectation over elements of $x$ and $\|x\|_\infty = \max\{|x_1|, |x_2|, \cdots, |x_{|\mathcal{S}\times\mathcal{A}|}|\}$, and, in *(c)*, we apply the conditions that $\gamma < 1$ and $x$ is a non-zero vector. Therefore, the null space of the matrix $I_{|\mathcal{S}\times\mathcal{A}|} - \gamma P^\pi$ only contains the zero vector, implying it is invertible. Thus, we can compute the successor measure by matrix inversion:

$$M^\pi = (1-\gamma)(I_{|\mathcal{S}\times\mathcal{A}|} - \gamma P^\pi)^{-1}.$$

Since the matrix $I_{|\mathcal{S}\times\mathcal{A}|} - \gamma P^\pi$ is invertible, we conclude that the successor measure is also a full-rank invertible matrix with rank$(M^\pi) = |\mathcal{S} \times \mathcal{A}|$. □

### B.2 Components for Unsupervised Pre-Training in RL

For *Step 1*, prior methods usually define a latent variable $z \in \mathcal{Z}$ sampled from a prior distribution $p_\mathcal{Z} \in \Delta(\mathcal{Z})$, e.g., a standard Gaussian distribution [30] or a scaled von Mises-Fisher distribution [81, 103], and use it to index latent-conditioned policies $\pi : \mathcal{S} \times \mathcal{Z} \to \Delta(\mathcal{A})$. The goal of unsupervised pre-training is to prefetch the reward-maximizing policies for downstream tasks [81, 103, 3, 5]. For *Step 2*, we are presented with a reward function $r(s, a)$ and asked to find a latent variable $z_r$ so that policy $\pi(a \mid s, z_r)$ achieves high reward.

### B.3 Definition of Ground-Truth Forward-Backward Representations

We formally define the ground-truth forward-backward representations $F^\star : \mathcal{S} \times \mathcal{A} \times \mathcal{Z} \to \mathcal{Z}$ and $B^\star : \mathcal{S} \times \mathcal{A} \to \mathcal{Z}$ as a pair of functions satisfying the following properties.

**Definition 2** (Definition 1 of Touati and Ollivier [103]). *For any CMP with latent space $\mathcal{Z}$ and any marginal probability measure $\rho \in \Delta(\mathcal{S}\times\mathcal{A})$, we say that a pair of functions $F^\star : \mathcal{S}\times\mathcal{A}\times\mathcal{Z} \to \mathcal{Z}$ and $B^\star : \mathcal{S} \times \mathcal{A} \to \mathcal{Z}$ are the ground-truth forward-backward representations if, for any latent variable $z$, any current state-action pair $(s, a)$, and any future state-action pair $(s_f, a_f)$, the latent-conditioned policy $\pi : \mathcal{S} \times \mathcal{Z} \to \Delta(\mathcal{A})$ defined by*

$$\pi(a \mid s, z) = \delta\big(a \,\big|\, \arg\max_{a\in\mathcal{A}} F^\star(s, a, z)^\top z\big), \tag{11}$$

*has its associated successor measure ratio $M^\pi/\rho : \mathcal{S} \times \mathcal{A} \times \mathcal{Z} \times \mathcal{S} \times \mathcal{A} \to \mathbb{R}_{\geq 0}$ satisfy*

$$\frac{M^\pi(s_f, a_f \mid s, a, z)}{\rho(s_f, a_f)} = F^\star(s, a, z)^\top B^\star(s_f, a_f). \tag{12}$$

**Definition 3** (heorem 2 of Touati and Ollivier [103]). *Augmenting the CMP with a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, for any latent variable $z$, and any current state-action pair $(s, a)$, the ground-truth FB representations $F^\star : \mathcal{S} \times \mathcal{A} \times \mathcal{Z} \to \mathcal{Z}$ and $B^\star : \mathcal{S} \times \mathcal{A} \to \mathcal{Z}$ produces a latent variable*

$$z_r = \mathbb{E}_{(s_f, a_f) \sim \rho(s_f, a_f)} \left[ B^\star(s_f, a_f) r(s_f, a_f) \right] \tag{13}$$

*that indexes the optimal policy $\pi_r^\star(a \mid s) = \pi(a \mid s, z_r)$ and the optimal Q-value $Q_r^\star(s, a) = F^\star(s, a, z_r)^\top z_r$.*

The optimal policy adaptation for any reward function depends on the closeness of the latent space $\mathcal{Z}$. In practice, prior work [103, 104, 5] usually sets the latent space to be the entire $d$-dimensional real space $\mathcal{Z} = \mathbb{R}^d$. One intriguing property of this design decision is the linear closeness of $\mathbb{R}^d$: $\mathbb{R}^d$ is a vector space that is closed under vector addition and scalar multiplication.

**Lemma 2** (Closeness of the $d$-dimensional real space). *For any vectors $x, y \in \mathbb{R}^d$ and any scalars $a, b \in \mathbb{R}$, the linear combination $ax + by \in \mathbb{R}^d$.*

Importantly, we can apply Lemma 2 to the (infinite number of) backward representations $B(s_f, a_f)$ and extend the results to an expectation (integral):

**Corollary 3** (The latent space covers the optimal latent variable for any reward function). *For a latent space $\mathcal{Z} = \mathbb{R}^d$, the ground-truth backward representations $B^\star : \mathcal{S} \times \mathcal{A} \to \mathcal{Z}$, any reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and any marginal probability measure $\rho \in \Delta(\mathcal{S} \times \mathcal{A})$, the optimal latent variable*

$$z_r = \mathbb{E}_{(s_f, a_f) \sim \rho(s_f, a_f)} \left[ B^\star(s_f, a_f) r(s_f, a_f) \right]$$

*is always covered by the latent space: $z_r \in \mathcal{Z}$.*

This corollary enables the FB algorithm to first pre-train a set of latent-conditioned policies and then use the optimal latent variable to index the optimal policy for any reward function.

# C Theoretical Analysis

## C.1 Existence of the Ground-Truth FB Representations

Before proving the existence of FB representations, we define some special notations for the latent space and use matrices to simplify our derivations. Specifically, we will consider the latent space $\mathcal{Z} = \mathbb{R}^d$ as both a latent manifold and a set of latent variables containing every vector in $\mathbb{R}^d$: $\mathcal{Z} = \{z_1, \cdots, z_{|\mathcal{Z}|}\}$. Although the size of the latent space is infinite, we will consider $|\mathcal{Z}|$ as a finite number and take the limit to infinity ($|\mathcal{Z}| \to \infty$). We will only use this notation to simplify our theoretical analysis.

For any forward representation function $F : \mathcal{S} \times \mathcal{A} \times \mathcal{Z} \to \mathcal{Z}$ and any backward representation function $B : \mathcal{S} \times \mathcal{A} \to \mathcal{Z}$, we stack the backward representations $B(s_f, a_f)$ into a matrix $B \in \mathbb{R}^{d \times |\mathcal{S} \times \mathcal{A}|}$:

$$B = \begin{bmatrix} B(s_1, a_1) & \cdots & B(s_{|\mathcal{S}|}, a_1) & \cdots & B(s_{|\mathcal{S}|}, a_{|\mathcal{A}|}) \end{bmatrix}. \tag{14}$$

The forward representations induce the latent-conditioned policy $\pi : \mathcal{S} \times \mathcal{Z} \to \Delta(\mathcal{A})$ (Eq. 11). These policies maximize the inner products between the forward representation and the corresponding latent variable: a delta measure (indicator function) around the maximizer.

$$\begin{aligned} \pi(a \mid s, z) &= \delta\left( a \,\middle|\, \arg\max_{a \in \mathcal{A}} F(s, a, z)^\top z \right) \\ &= \mathbb{1}_{\arg\max_{a \in \mathcal{A}} F(s,a,z)^\top z}(a). \end{aligned} \tag{15}$$

For different latent $z_i \in \mathcal{Z}$, the policy $\pi(a \mid s, z_i)$ induces a successor measure matrix $M_i^\pi \in \mathbb{R}^{|\mathcal{S} \times \mathcal{A}| \times |\mathcal{S} \times \mathcal{A}|}$ as computed in Lemma 1: for all $i = 1, \cdots, |\mathcal{Z}|$,

$$M_i^\pi = (1 - \gamma) \left( I_{|\mathcal{S} \times \mathcal{A}|} - \gamma P^{\pi(a|s, z_i)} \right)^{-1}. \tag{16}$$

We also aggregate all the $M_i^\pi$'s into a single matrix $M_{\mathcal{Z}}^\pi \in \mathbb{R}^{|\mathcal{Z}\times\mathcal{S}\times\mathcal{A}|\times|\mathcal{S}\times\mathcal{A}|}$:

$$M_{\mathcal{Z}}^\pi = \begin{bmatrix} M_1^\pi \\ \vdots \\ M_{|\mathcal{Z}|}^\pi \end{bmatrix}. \tag{17}$$

Similarly, for each $z_i$, we stack the latent-conditioned forward representations $F(s, a, z_i)$ into a matrix $F_i \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times d}$: for $i = 1, \cdots, |\mathcal{Z}|$,

$$F_i = \begin{bmatrix} F(s_1, a_1, z_i)^\top \\ \vdots \\ F(s_1, a_{|\mathcal{A}|}, z_i)^\top \\ \vdots \\ F(s_{|\mathcal{S}|}, a_{|\mathcal{A}|}, z_i)^\top \end{bmatrix}, \tag{18}$$

and also aggregate all the $F_i$'s into a single matrix $F_{\mathcal{Z}}^\pi \in \mathbb{R}^{|\mathcal{Z}\times\mathcal{S}\times\mathcal{A}|\times d}$:

$$F_{\mathcal{Z}} = \begin{bmatrix} F(s_1, a_1, z_1)^\top \\ \vdots \\ F(s_1, a_{|\mathcal{A}|}, z_1)^\top \\ \vdots \\ F(s_{|\mathcal{S}|}, a_{|\mathcal{A}|}, z_1)^\top \\ \vdots \\ F(s_{|\mathcal{S}|}, a_{|\mathcal{A}|}, z_{|\mathcal{Z}|})^\top \end{bmatrix} = \begin{bmatrix} F_1 \\ \vdots \\ F_{|\mathcal{Z}|} \end{bmatrix}. \tag{19}$$

Finally, given a marginal probability measure (probability mass) $\rho \in \Delta(\mathcal{S} \times \mathcal{A})$ with full support on $\mathcal{S} \times \mathcal{A}$ and a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, with slight abuse of notation, we stack them in a marginal measure vector $\rho \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ and a reward vector $r \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$, respectively:

$$\rho = \begin{bmatrix} \rho\left(s_1, a_1\right) \\ \vdots \\ \rho\left(s_1, a_{|\mathcal{A}|}\right) \\ \vdots \\ \rho\left(s_{|\mathcal{S}|}, a_{|\mathcal{A}|}\right) \end{bmatrix}, \qquad r = \begin{bmatrix} r\left(s_1, a_1\right) \\ \vdots \\ r\left(s_1, a_{|\mathcal{A}|}\right) \\ \vdots \\ r\left(s_{|\mathcal{S}|}, a_{|\mathcal{A}|}\right) \end{bmatrix}. \tag{20}$$

Defining these matrices allows us to simplify the notation and denote the FB representation learning procedure using linear algebra. For example, the successor measure ratio identity in Definition 2 and the optimal latent adaptation in Definition 3 can be written as

$$M_{\mathcal{Z}}^\pi \mathrm{diag}(\rho)^{-1} = F_{\mathcal{Z}}^\star B^\star, \qquad z_r = B^\star(r \odot \rho),$$

where $\mathrm{diag}(\rho) \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times|\mathcal{S}\times\mathcal{A}|}$ is the diagonal matrix of the marginal measure vector $\rho$ and $\odot$ denotes the element-wise multiplication. In addition, by the relationship between the successor measure and the Q-value (Eq. 2), we can write the Q-value for a latent-conditioned policy $\pi(a \mid s, z_i)$ as a vector in $\mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$: $Q_i^\pi = M_i^\pi r$.

We can now constrain the rank of the forward representation matrix $\mathrm{rank}\,(F_{\mathcal{Z}})$, the rank of the backward representation matrix $\mathrm{rank}(B)$, and the rank of the product of the forward-backward representation matrices $\mathrm{rank}\,(F_{\mathcal{Z}}B)$ by matrix dimensions.

**Remark 1.** *The rank of any forward representation matrix satisfies* $\mathit{rank}(F_{\mathcal{Z}}) \leq \mathit{min}\,(|\mathcal{Z} \times \mathcal{S} \times \mathcal{A}|, d)$. *The rank of any backward representation matrix satisfies* $\mathit{rank}(B) \leq \mathit{min}\,(d, |\mathcal{S} \times \mathcal{A}|)$. *The rank of the product of the forward-backward representation matrices satisfies* $\mathit{rank}\,(F_{\mathcal{Z}}B) \leq \mathit{min}\,(\mathit{rank}(F_{\mathcal{Z}}), \mathit{rank}(B))$.

Using these constraints, we formally prove the existence of FB representations.

**Proposition 1.** *Given any discrete CMP, a latent space $\mathcal{Z} = \{z_1, \cdots, z_{|\mathcal{Z}|}\}$ with each $z_i \in \mathbb{R}^d$, and any marginal probability measure vector $\rho \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ (Eq. 20), any forward representation matrix $F^\star_{\mathcal{Z}} \in \mathbb{R}^{|\mathcal{Z}\times\mathcal{S}\times\mathcal{A}|\times d}$ (Eq. 19), which induces the latent-conditioned policy $\pi : \mathcal{S}\times\mathcal{Z} \to \Delta(\mathcal{A})$ (Eq. 15) with associated successor measure matrix $M^\pi_{\mathcal{Z}}$ (Eq. 17), and any backward representation matrix $B^\star \in \mathbb{R}^{d\times|\mathcal{S}\times\mathcal{A}|}$ (Eq. 14) that encodes this CMP's successor measure as,*

1. *$F^\star_{\mathcal{Z}}$ and $B^\star$ fit the successor measure ratio (Definition 2):*

$$M^\pi_{\mathcal{Z}} diag(\rho)^{-1} = F^\star_{\mathcal{Z}} B^\star, \tag{21}$$

2. *$F^\star_{\mathcal{Z}}$ and $B^\star$ enable optimal policy adaptation (Definition 3): for any reward vector $r \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ (Eq. 20),*

$$z_r = B^\star(r \odot \rho) \in \mathcal{Z}, \tag{22}$$

*indexes the optimal policy $\pi^\star_r(a \mid s) = \pi(a \mid s, z_r) = \arg\max_a F^\star(s, a, z_r)^\top z_r$ and the optimal Q-value $Q^\star_r = Q_{z_r} = F^\star_{z_r} z_r$,*

*must satisfy the following properties:*

1. *The representation dimension $d$ is at least $|\mathcal{S}\times\mathcal{A}|$, i.e., $d \geq |\mathcal{S}\times\mathcal{A}|$.*
2. *The rank of the forward representation matrix $F^\star_{\mathcal{Z}}$ is at least $|\mathcal{S}\times\mathcal{A}|$ and at most $d$, i.e., $|\mathcal{S}\times\mathcal{A}| \leq rank\,(F^\star_{\mathcal{Z}}) \leq d$.*
3. *The rank of the backward representation matrix $B^\star$ is equivalent to $|\mathcal{S}\times\mathcal{A}|$, i.e., $rank\,(B^\star) = |\mathcal{S}\times\mathcal{A}|$.*
4. *For different latents $z_i (i = 1, \cdots, |\mathcal{Z}|)$, the backward representation matrix $B^\star$, the forward representation matrix for each latent $F^\star_i$ (Eq. 18), and the successor measure matrix for each latent $M^\pi_i$ (Eq. 16) must satisfy:*

$$B^\star = F^{\star+}_1 M^\pi_1 diag(\rho)^{-1} = F^{\star+}_2 M^\pi_2 diag(\rho)^{-1} = \cdots = F^{\star+}_{|\mathcal{Z}|} M^\pi_{|\mathcal{Z}|} diag(\rho)^{-1},$$

*where $X^+$ denotes the pseudoinverse (Moore–Penrose inverse) [69, 11] of the matrix $X$ and $diag(x)$ is the diagonal matrix of the vector $x$.*

*Proof.* The main idea of our proof is to match the rank of the FB representation matrices $F^\star_{\mathcal{Z}}, B^\star$ to the rank of the successor measure matrix $M^\pi_{\mathcal{Z}}$.

**Rank matching.** Since we aim to find a forward representation matrix $F^\star_{\mathcal{Z}}$ and a backward representation matrix $B^\star$ fitting the successor measure ratio exactly, it is necessary to match the rank on both sides of Eq. 21:

$$\text{rank}\,(M^\pi_{\mathcal{Z}}) = \text{rank}\left(M^\pi_{\mathcal{Z}} \text{diag}(\rho)^{-1}\right) = \text{rank}(F^\star_{\mathcal{Z}} B^\star).$$

Applying Lemma 1 to the successor measure of each latent $z_i \in \mathcal{Z}$ gives us $\text{rank}(M^\pi_i) = |\mathcal{S}\times\mathcal{A}|$. Meanwhile, because the successor measure matrix $M^\pi_{\mathcal{Z}}$ is an aggregation of each $M^\pi_i$ (Eq. 17), we have $\text{rank}(M^\pi_{\mathcal{Z}}) = |\mathcal{S}\times\mathcal{A}|$, suggesting that the rank of the product of the FB representation matrices must be equivalent to $|\mathcal{S}\times\mathcal{A}|$:

$$\text{rank}(M^\pi_{\mathcal{Z}}) = \text{rank}\,(F^\star_{\mathcal{Z}} B^\star) = |\mathcal{S}\times\mathcal{A}|.$$

We next constrain the rank of the forward representation matrix $F^\star_{\mathcal{Z}}$ and the rank of the backward representation matrix $B^\star$ by using the properties in Remark 1. Since the rank of the product of the forward-backward representation matrices is at most the rank of either representation matrix, we have

$$\begin{aligned} &\text{rank}\,(F^\star_{\mathcal{Z}} B^\star) \leq \min\left(\text{rank}\,(F^\star_{\mathcal{Z}}), \text{rank}\,(B^\star)\right) \\ \implies &\text{rank}\,(F^\star_{\mathcal{Z}} B^\star) \leq \text{rank}\,(F^\star_{\mathcal{Z}}) \quad \text{and} \quad \text{rank}\,(F^\star_{\mathcal{Z}} B^\star) \leq \text{rank}\,(B^\star). \end{aligned}$$

Plugging in the rank of the product of FB representation matrices $\text{rank}(F_{\mathcal{Z}}^{\star}B^{\star}) = |\mathcal{S} \times \mathcal{A}|$ and the constraints for the rank of the forward representation matrix $\text{rank}(F_{\mathcal{Z}}^{\star})$ and the rank of the backward representation matrix $\text{rank}(B^{\star})$, we have

$$\begin{aligned} & |\mathcal{S} \times \mathcal{A}| \leq \text{rank}(F_{\mathcal{Z}}^{\star}) \leq \min\left(|\mathcal{Z} \times \mathcal{S} \times \mathcal{A}|, d\right) \quad \text{and} \quad |\mathcal{S} \times \mathcal{A}| \leq \text{rank}(B^{\star}) \leq \min\left(d, |\mathcal{S} \times \mathcal{A}|\right) \\ \implies & |\mathcal{S} \times \mathcal{A}| \leq \text{rank}(F_{\mathcal{Z}}^{\star}) \leq \min\left(|\mathcal{Z} \times \mathcal{S} \times \mathcal{A}|, d\right) \quad \text{and} \quad |\mathcal{S} \times \mathcal{A}| \leq d,\ \text{rank}(B^{\star}) = |\mathcal{S} \times \mathcal{A}| \\ & \overset{(a)}{\implies} |\mathcal{S} \times \mathcal{A}| \leq \text{rank}(F_{\mathcal{Z}}^{\star}) \leq d \quad \text{and} \quad |\mathcal{S} \times \mathcal{A}| \leq d,\ \text{rank}(B^{\star}) = |\mathcal{S} \times \mathcal{A}|, \end{aligned} \tag{23}$$

where we simplify the inequalities in *(a)* when $|\mathcal{Z}| \to \infty$.

These results suggest that the rank constraints on the FB representation matrices:

- The rank of the forward representation matrix is at least $|\mathcal{S} \times \mathcal{A}|$ (not necessarily full rank).
- The rank of the backward representation matrix is equivalent to $|\mathcal{S} \times \mathcal{A}|$ (*full* rank).

In addition, the necessary condition for the existence of ground-truth FB representation matrices is that the representation dimension is at least $|\mathcal{S} \times \mathcal{A}|$, i.e., $d \geq |\mathcal{S} \times \mathcal{A}|$. Importantly, these are three individual conditions for the representation dimension $d$, the forward representation matrix $F_{\mathcal{Z}}^{\star}$, and the backward representation matrix $B^{\star}$, respectively. However, they still fail to guarantee that the product of the FB representation matrices $F_{\mathcal{Z}}^{\star}B$ will fit $M_{\mathcal{Z}}^{\pi}\text{diag}(\rho)^{-1}$ exactly. We need further relationships to bridge $F_{\mathcal{Z}}^{\star}$ and $B^{\star}$.

**Bridging the FB representation matrices.** Our key observations are twofold. *First*, the backward representation matrix $B^{\star}$ does not take any latent variable as input, indicating that $B$ compresses the common information throughout the entire latent space. *Second*, the rank matching not only holds for the entire $F_{\mathcal{Z}}^{\star}$ and $M_{\mathcal{Z}}^{\star}$ matrices, but also holds for the forward representation matrix of each latent $F_i^{\star}$ (Eq. 16) and the successor measure ratio of each latent $M_i^{\pi}$ (Eq. 18). We next discuss the meaning of these two observations.

When the ground-truth FB representation matrices exist, using the block matrix notations in Eq. 17 and Eq. 19 to rewrite Eq. 21 gives us

$$\begin{aligned} & \begin{bmatrix} M_1^{\pi} \\ \vdots \\ M_{|\mathcal{Z}|}^{\pi} \end{bmatrix} \text{diag}(\rho)^{-1} = \begin{bmatrix} F_1^{\star} \\ \vdots \\ F_{|\mathcal{Z}|}^{\star} \end{bmatrix} B^{\star} \\ \implies & M_1^{\pi}\text{diag}(\rho)^{-1} = F_1^{\star}B^{\star},\ M_2^{\pi}\text{diag}(\rho)^{-1} = F_2^{\star}B^{\star},\ \cdots,\ M_{|\mathcal{Z}|}^{\pi}\text{diag}(\rho)^{-1} = F_{|\mathcal{Z}|}^{\star}B^{\star}. \end{aligned}$$

Furthermore, since, for $i = 1, \cdots, |\mathcal{Z}|$, each successor measure $M_i^{\pi} \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times|\mathcal{S}\times\mathcal{A}|}$ is a square matrix with rank $\text{rank}(M_i^{\pi}) = |\mathcal{S} \times \mathcal{A}|$, the column space of each successor measure is equivalent to the $|\mathcal{S} \times \mathcal{A}|$-dimensional real space:

$$\text{col}\left(M_i^{\pi}\right) = \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}.$$

We note that the marginal probability measure vector $\rho$ does not change the column space of $M_i^{\pi}$ because $\rho$ has the full support over $\mathcal{S} \times \mathcal{A}$: $\text{col}\left(M_i^{\pi}\text{diag}(\rho)^{-1}\right) = \text{col}\left(M_i^{\pi}\right)$. Meanwhile, since the rank of the entire forward representation matrix $\text{rank}(F_{\mathcal{Z}}^{\star})$ is at least $|\mathcal{S} \times \mathcal{A}|$, we know that the rank of each $\text{rank}(F_i^{\star})$ is also at least $|\mathcal{S} \times \mathcal{A}|$. By the shape of matrix $F_i^{\star}$, this observation indicates that, for $i = 1, \cdots, |\mathcal{Z}|$,

$$|\mathcal{S} \times \mathcal{A}| \leq \text{rank}(F_i^{\star}) \leq \min\left(|\mathcal{S} \times \mathcal{A}|, d\right) \quad \text{and} \quad d \geq |\mathcal{S} \times \mathcal{A}| \implies \text{rank}(F_i^{\star}) = |\mathcal{S} \times \mathcal{A}|.$$

Thus, the column space of each forward representation matrix $F_i^{\star}$ is also equivalent to the $|\mathcal{S} \times \mathcal{A}|$-dimensional real space:

$$\text{col}(F_i^{\star}) = \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}.$$

Therefore, by the definition of the pseudoinverse of a matrix, we have

$$\text{col}\left(M_i^{\pi}\text{diag}(\rho)^{-1}\right) = \text{col}\left(M_i^{\pi}\right) = \text{col}(F_i^{\star}) \implies F_i^{\star}F_i^{\star+}M_i^{\pi}\text{diag}(\rho)^{-1} = M_i^{\pi}\text{diag}(\rho)^{-1},$$

where $F_i^{\star +}$ is the pseudoinverse of matrix $F_i^\star$.

These intriguing observations help us find the additional conditions for the existence of ground-truth FB representation matrices: the ground-truth backward representation matrix $B^\star$ must be shared by each forward representation matrix $F_i^\star$ and each successor measure matrix $M_i^\pi$ as

$$B^\star = F_1^{\star +} M_1^\pi \text{diag}(\rho)^{-1} = F_2^{\star +} M_2^\pi \text{diag}(\rho)^{-1} = \cdots = F_{|\mathcal{Z}|}^{\star +} M_{|\mathcal{Z}|}^\pi \text{diag}(\rho)^{-1}.$$

Importantly, the ground-truth FB representation matrices are not unique because we can multiply both $F_i^\star$ and $B^\star$ by the same orthonormal rotation matrix $Q_{\text{rot}}$ to recover the same product.

**Verifying the optimal policy adaptation.** Until now, we have only focused on discussing the conditions for the existence of ground-truth FB representation matrices $F_{\mathcal{Z}}^\star, B$ that fit the successor measure ratio $M_{\mathcal{Z}}^\pi \text{diag}(\rho)^{-1}$. It remains unclear whether this pair of FB representation matrices will enable optimal policy adaptation (Eq. 22). We next prove that the latent variable $z_r$ recovers the optimal policy and the optimal Q-value for any reward vector $r \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$.

Formally, given the FB representation matrices $F_{\mathcal{Z}}^\star, B$ and the reward vector $r$, we denote the forward representation matrix for the latent variable $z_r$ as $F_{z_r} \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times|\mathcal{S}\times\mathcal{A}|}$ (an example of Eq. 18). This forward representation matrix $F_{z_r}^\star$ induces a latent-conditioned policy $\pi(a \mid s, z_r)$ with the associated successor measure matrix $M_{z_r}^\pi \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times|\mathcal{S}\times\mathcal{A}|}$. Together with the latent variable $z_r$, the forward representation matrix $F_{z_r}^\star$ and the successor measure matrix $M_{z_r}^\pi$ satisfy

$$\begin{aligned} F_{z_r}^\star z_r &= F_{z_r}^\star B^\star (r \odot \rho) \\ &\overset{(a)}{=} M_{z_r}^\pi \text{diag}(\rho)^{-1} (r \odot \rho) \\ &= M_{z_r}^\pi r \\ &\overset{(b)}{=} Q_{z_r}, \end{aligned}$$

where, in *(a)*, we apply the definition of the inverse of a diagonal matrix and the definition of elementwise product, and, in *(b)*, we apply the definition of the Q-value vector. Since the latent-conditioned policy $\pi(a \mid s, z_r)$ is maximizing the inner product $F^\star(s, a, z_r)^\top z_r$ (Eq. 15), we have $\pi(a \mid s, z_r) = \arg\max Q_{z_r}(s, a)$. By definition, $Q_{z_r}$ is the optimal Q-value vector for the reward vector $r$ with the optimal policy $\pi(a \mid s, z_r)$. □

### C.2 Low-Rank Approximation Incurs Arbitrary Errors

**Corollary 2.** *Given any marginal probability measure vector $\rho \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ and a representation dimension $d < |\mathcal{S} \times \mathcal{A}|$, the FB representation matrices $F_{\mathcal{Z}} \in \mathbb{R}^{|\mathcal{Z}\times\mathcal{S}\times\mathcal{A}|\times d}$ and $B \in \mathbb{R}^{d\times|\mathcal{S}\times\mathcal{A}|}$ learned by the FB algorithm is a low-rank approximation of the successor measure ratio $M_{\mathcal{Z}}^\pi diag(\rho)^{-1} \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$.*

*For any reward vector $r \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$, this low-rank approximation induces the latent variable $z_r = B(r \odot \rho) \in \mathbb{R}^d$ and the optimal Q-value prediction $Q_{z_r} = F_{z_r} z_r \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$. Denote the error in the optimal Q-value prediction as*

$$\epsilon(r) = \|Q_r^\star - F_{z_r} z_r\|_\infty.$$

*Then, for any $c > 0$, there exists a reward vector $r_{null} \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ in the null space of $B$ such that $\epsilon(r_{null}) \geq c$.*

*Proof.* From Proposition 1, we know that the rank of the successor measure matrix $M_{\mathcal{Z}}^\pi$ is $|\mathcal{S} \times \mathcal{A}|$. When $d < |\mathcal{S} \times \mathcal{A}|$, the FB representation matrices produce a low-rank approximation on the successor measure ratio:

$$\text{rank}(F_{\mathcal{Z}} B) \leq d < |\mathcal{S} \times \mathcal{A}| = \text{rank}(M_{\mathcal{Z}}^\pi \text{diag}(\rho)^{-1}).$$

In this case, the rank of the backward representation matrix satisfies

$$\text{rank}(B) \leq \min(d, |\mathcal{S} \times \mathcal{A}|) = d < |\mathcal{S} \times \mathcal{A}|.$$

Thus, the backward representation matrix is *not* full column rank, suggesting that there exists a *non-zero* reward vector $r_{\text{null}} \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ in the null space of $B$ that induces a zero latent variable:

$$z_{r_{\text{null}}} = B(r_{\text{null}} \odot \rho) = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^d.$$

For this zero latent variable, we always have $F_{z_{r_{\text{null}}}} z_{r_{\text{null}}} = 0$. However, the optimal Q-value for the reward $r_{\text{null}}$ is not necessarily zero. Therefore, the optimal Q-value prediction error $\epsilon(r_{\text{null}})$ can be arbitrarily large by scaling the non-zero entries in $r_{\text{null}}$. We conclude that for any $c > 0$, there exists a reward vector $r_{\text{null}}$ such that $\epsilon(r_{\text{null}}) \geq c$. In other words, the errors in the optimal Q-value prediction can be arbitrarily large. □

### C.3 Equivariant to Affine Transformations of Rewards

We first prove the equivariant property of the Q-value for any positive affine transition under any policy. We will then use this property to derive the equivariant property of the ground-truth forward representations in FB.

**Lemma 3.** *Given a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, a positive scalar $\nu > 0$, and an offset $\xi \in \mathbb{R}$, for a state-action pair $(s, a)$, let the transformed reward be $\nu r(s, a) + \xi$. Then, the Q-value of the reward function $r$ and the Q-value of the reward $\nu r + \xi$ satisfies $Q^\pi_{\nu r+\xi}(s, a) = \nu Q^\pi_r(s, a) + \xi$.*

The one-line proof of this lemma will use the definition of Q-value, which is a sum of cumulative discounted rewards, and apply the affine transformation to each reward in that summation. Now, for the ground-truth forward representations, we also have the equivariant property:

**Proposition 2.** *For a state-action pair $(s, a)$, a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, a positive scalar $\nu > 0$, an offset $\xi \in \mathbb{R}$, the ground-truth FB representation functions $F^\star : \mathcal{S} \times \mathcal{A} \times \mathcal{Z} \to \mathcal{Z}$, $B^\star : \mathcal{S} \times \mathcal{A} \to \mathcal{Z}$, and a marginal measure $\rho \in \Delta(\mathcal{S} \times \mathcal{A})$, let the transformed reward function be $\nu r(s, a) + \xi$. Then, $z_r$ is the latent variable indexing the optimal Q-value for the reward $r$, and $z_{\nu r+\xi}$ is the latent variable indexing the optimal Q-value for the reward $\nu r + \xi$. Furthermore, the ground-truth forward-backward representations are invariant to the affine transformation with positive scaling in latent variables, i.e., $F^\star(s, a, z_{\nu r+\xi}) = F^\star(s, a, z_r)$.*

*Proof.* By Eq. 13 in Definition 3, we can write $z_r$ and $z_{\nu r+\xi}$ as

$$\begin{aligned}
z_r &= \int_{\mathcal{S}\times\mathcal{A}} B^\star(s_f, a_f) r(s_f, a_f) \rho(s_f, a_f) ds_f da_f \\
z_{\nu r+\xi} &= \int_{\mathcal{S}\times\mathcal{A}} B^\star(s_f, a_f)(\nu r(s_f, a_f) + \xi) \rho(s_f, a_f) ds_f da_f \\
&= \nu \int_{\mathcal{S}\times\mathcal{A}} B^\star(s_f, a_f) r(s_f, a_f) \rho(s_f, a_f) ds_f da_f + \xi \int_{\mathcal{S}\times\mathcal{A}} B^\star(s_f, a_f) \rho(s_f, a_f) ds_f da_f \\
&= \nu z_r + \xi z_{\text{one}},
\end{aligned} \tag{24}$$

where we denote the latent variable for a reward that consistently equals 1 ($r(s, a) = 1$) as $z_{\text{one}}$. One intriguing property of the latent variable $z_{\text{one}}$ is that, for any other latent variable $z$

$$\begin{aligned}
F^\star(s, a, z)^\top z_{\text{one}} &= \int_{\mathcal{S}\times\mathcal{A}} F^\star(s, a, z)^\top B^\star(s_f, a_f) \rho(s_f, a_f) ds_f da_f \\
&\overset{(a)}{=} \int_{\mathcal{S}\times\mathcal{A}} \frac{M^\pi(s_f, a_f \mid s, a, z)}{\rho(s_f, a_f)} \cdot \rho(s_f, a_f) ds_f da_f \\
&= \int_{\mathcal{S}\times\mathcal{A}} M^\pi(s_f, a_f \mid s, a, z) ds_f da_f \\
&\overset{(b)}{=} 1,
\end{aligned} \tag{25}$$

where, in *(a)*, we use the definition of ground-truth FB representations in Eq. 12, and, in *(b)*, we apply the definition of the successor measure. Since $z_r$ indexes the optimal Q-value for the reward $r$ and $z_{\nu r+\xi}$ indexes the optimal Q-value for the reward $\nu r + \xi$, we have

$$
\begin{aligned}
Q^\star_r(s,a) &= F^\star(s,a,z_r)^\top z_r \\
Q^\star_{\nu r+\xi}(s,a) &= F^\star(s,a,z_{\nu r+\xi})^\top z_{\nu r+\xi} \\
&\overset{(a)}{=} \nu F^\star(s,a,\nu z_r + \xi z_{\text{one}})^\top z_r + \xi F^\star(s,a,\nu z_r + \xi z_{\text{one}})^\top z_{\text{one}} \\
&\overset{(b)}{=} \nu F^\star(s,a,z_{\nu r+\xi})^\top z_r + \xi,
\end{aligned}
$$

where, in *(a)*, we apply the relationship in Eq. 24, and, in *(b)*, we apply the property of $z_{\text{one}}$ in Eq. 25. Finally, since an affine transformation with positive scaling does not change the optimal policy [91, 74], the Q-value $Q^\star_r$ and the Q-value $Q^\star_{\nu r+\xi}$ satisfy Lemma 3. Using the conclusion from Lemma 3, we have

$$
\begin{aligned}
Q^\star_{\nu r+\xi}(s,a) &= \nu Q^\star_r(s,a) + \xi \\
\implies \nu F^\star(s,a,z_{\nu r+\xi})^\top z_r + \xi &= \nu F^\star(s,a,z_r)^\top z_r + \xi \\
\implies F^\star(s,a,z_{\nu r+\xi}) &= F^\star(s,a,z_r),
\end{aligned}
$$

where the last identity holds because $\nu > 0$ and $\xi \in \mathbb{R}$ are arbitrary. □

### C.4 Deriving the FB Representation Learning Objective

We derive the LSIF loss for the FB algorithm that learns forward-backward representations in a temporal-difference manner. First, we replace the successor measure in Eq. 4 using the recursive Bellman equation in Eq. 1, decomposing the ratio $M^\pi(s_f, a_f \mid s, a, z)/\rho(s_f, a_f)$ into a convex combination (with weight $\gamma$) between the in-place ratio $\delta(s_f, a_f \mid s, a)/\rho(s_f, a_f)$ and the ratio at the next time step $M^\pi(s_f, a_f \mid s', a', z)/\rho(s_f, a_f)$:

$$
\frac{1}{2}\mathbb{E}_{\substack{p(s,a,z),\,\rho(s_f,a_f)\\ p(s'|s,a),\,\pi(a'|s',z)}}\left[\left(F(s,a,z)^\top B(s_f,a_f) - (1-\gamma)\frac{\delta(s_f,a_f \mid s,a)}{\rho(s_f,a_f)} - \gamma\frac{M^\pi(s_f,a_f \mid s',a',z)}{\rho(s_f,a_f)}\right)^2\right].
$$

Second, we use target forward-backward representation functions $\bar{F}$ and $\bar{B}$ to replace the ground-truth ratio at the next time step $M^\pi(s_f, a_f \mid s', a', z)/\rho(s_f, a_f)$. The resulting loss function minimizes a Bellman error:

$$
\begin{aligned}
\mathcal{L}_{\text{TD FB}}(F,B) &= \frac{1}{2}\mathbb{E}_{\substack{p(s,a,z),\,\rho(s_f,a_f)\\ p(s'|s,a),\,\pi(a'|s',z)}}\left[\left(F(s,a,z)^\top B(s_f,a_f) - y\right)^2\right], \\
y &= (1-\gamma)\frac{\delta(s_f,a_f \mid s,a)}{\rho(s_f,a_f)} + \gamma \bar{F}(s',a',z)^\top \bar{B}(s_f,a_f).
\end{aligned}
$$

Now, expanding the mean squared error gives us

$$
\begin{aligned}
\mathcal{L}_{\text{TD FB}}(F, B) &= \frac{1}{2}\mathbb{E}_{\substack{p(s,a,z),\, \rho(s_f,a_f) \\ p(s'|s,a),\, \pi(a'|s',z)}} \Big[ \big(F(s,a,z)^\top B(s_f,a_f)\big)^2 \\
&\quad -2 \cdot F(s,a,z)^\top B(s_f,a_f) \cdot \left( (1-\gamma) \frac{\delta(s_f, a_f \mid s,a)}{\rho(s_f,a_f)} + \gamma \bar{F}(s',a',z)^\top \bar{B}(s_f,a_f) \right) \Big] + \text{const.} \\
&= \frac{1}{2}\mathbb{E}_{p(s,a,z),\, \rho(s_f,a_f)} \Big[ \big(F(s,a,z)^\top B(s_f,a_f)\big)^2 \Big] \\
&\quad - (1-\gamma)\mathbb{E}_{p(s,a,z),\, \rho(s_f,a_f)} \left[ \frac{\delta(s_f, a_f \mid s,a)}{\rho(s_f,a_f)} F(s,a,z)^\top B(s_f,a_f) \right] \\
&\quad - \gamma \mathbb{E}_{\substack{p(s,a,z),\, \rho(s_f,a_f) \\ p(s'|s,a),\, \pi(a'|s',z)}} \left[ \bar{F}(s',a',z)^\top \bar{B}(s_f,a_f) \cdot F(s,a,z)^\top B(s_f,a_f) \right] + \text{const.} \\
&\overset{(a)}{=} \frac{1}{2}\mathbb{E}_{p(s,a,z),\, \rho(s_f,a_f)} \Big[ \big(F(s,a,z)^\top B(s_f,a_f)\big)^2 \Big] - (1-\gamma)\mathbb{E}_{p(s,a,z)} \left[ F(s,a,z)^\top B(s,a) \right] \\
&\quad - \gamma \mathbb{E}_{\substack{p(s,a,z),\, \rho(s_f,a_f) \\ p(s'|s,a),\, \pi(a'|s',z)}} \left[ \bar{F}(s',a',z)^\top \bar{B}(s_f,a_f) \cdot F(s,a,z)^\top B(s_f,a_f) \right] + \text{const.} \\
&\overset{(b)}{=} \frac{1}{2}\mathbb{E}_{\substack{p(s,a,z),\, \rho(s_f,a_f) \\ p(s'|s,a),\, \pi(a'|s',z)}} \Big[ \big(F(s,a,z)^\top B(s_f,a_f) - \gamma \bar{F}(s',a',z)^\top \bar{B}(s_f,a_f)\big)^2 \Big] \\
&\quad - (1-\gamma)\mathbb{E}_{p(s,a,z)} \left[ F(s,a,z)^\top B(s,a) \right] + \text{const.},
\end{aligned}
$$

where, in *(a)*, we apply the property of the delta measure, and, in *(b)*, we rearrange the quadratic terms by definition. Finally, we can ignore the constant for simplicity.

### C.5 The FB Bellman Operator Is Not a $\gamma$-Contraction

This section proves a negative result about the FB Bellman operator $\mathcal{T}_{\text{FB}}$. We show that the FB Bellman operator is not a $\gamma$-contraction. Therefore, the Banach fixed-point theorem fails to provide a guarantee to the FB algorithm that iteratively applies the FB Bellman operator to the forward-backward representation functions from the previous iteration. These results suggest that we need alternative theoretical tools to prove the convergence of the FB algorithm to a fixed point.

**Proposition 3.** *For any $\gamma \in [0,1)$ and $p \geq 1$, the FB Bellman operator $\mathcal{T}_{FB}$ is not a $\gamma$-contraction under the $L^p$-norm. Thus, the Banach fixed-point theorem is not applicable to the FB Bellman operator.*

*Proof.* We prove that the FB Bellman operator is not a $\gamma$-contraction by contradiction. Let the FB Bellman operator $\mathcal{T}_{\text{FB}}$ be a $\gamma$-contraction under the $L^p$-norm for any $\gamma \in [0,1)$ and $p \geq 1$. This indicates that for any two pairs of forward-backward representation functions $f_1 : \mathcal{S} \times \mathcal{A} \times \mathcal{Z} \to \mathcal{Z}$, $b_1 : \mathcal{S} \times \mathcal{A} \to \mathcal{Z}$, and $f_2 : \mathcal{S} \times \mathcal{A} \times \mathcal{Z} \to \mathcal{Z}$, $b_2 : \mathcal{S} \times \mathcal{A} \to \mathcal{Z}$, which induced the latent-conditioned policies $\pi_1(a \mid s, z) = \delta\left(a \mid \arg\max_{a \in \mathcal{A}} f_1(s,a,z)^\top z\right)$ and $\pi_2(a \mid s, z) = \delta\left(a \mid \arg\max_{a \in \mathcal{A}} f_2(s,a,z)^\top z\right)$, we have

$$
\left\| \mathcal{T}_{\text{FB}}(f_1^\top b_1) - \mathcal{T}_{\text{FB}}(f_2^\top b_2) \right\|_p \leq \gamma \left\| f_1^\top b_1 - f_2^\top b_2 \right\|_p. \tag{26}
$$

We consider any current state-action pair $(s,a)$, any latent $z \in \mathcal{Z}$, and any future state-action pair $(s_f, a_f)$. For the LHS of the inequality, we have

$$
\begin{aligned}
&\mathcal{T}_{\text{FB}}(f_1(s,a,z)^\top b_1(s_f,a_f)) - \mathcal{T}_{\text{FB}}(f_2(s,a,z)^\top b_2(s_f,a_f)) \\
&= \gamma \mathbb{E}_{p(s'|s,a),\pi_1(a'|s',z)} \left[ f_1(s',a',z)^\top b_1(s_f,a_f) \right] - \gamma \mathbb{E}_{p(s'|s,a),\pi_2(a'|s',z)} \left[ f_2(s',a',z)^\top b_2(s_f,a_f) \right] \\
&= \gamma \mathbb{E}_{p(s'|s,a)} \left[ \mathbb{E}_{\pi_1(a'|s',z)} \left[ f_1(s',a',z)^\top b_1(s_f,a_f) \right] - \mathbb{E}_{\pi_2(a'|s',z)} \left[ f_2(s',a',z)^\top b_2(s_f,a_f) \right] \right].
\end{aligned} \tag{27}
$$

Without loss of generality, we can set $f_2 = Q_{\text{rot}} f_1$ and $b_2 = Q_{\text{rot}} b_1$, where $Q_{\text{rot}} \in \mathbb{R}^{d \times d}$ is an orthonormal rotation matrix, as in Touati et al. [104]. In this case, the inner products satisfy $f_2(s,a,z)^\top b_2(s_f,a_f) = f_1(s,a,z)^\top Q_{\text{rot}}^\top Q_{\text{rot}} b_1(s_f,a_f) = f_1(s,a,z)^\top b_1(s_f,a_f)$. Thus, the RHS of Eq. 26 is always zero. However, the policy $\pi_1(a \mid s, z) = \delta\left(a \mid \arg\max_{a \in \mathcal{A}} f_1(s,a,z)^\top z\right)$

and the policy $\pi_2(a \mid s, z) = \delta\left(a \mid \arg\max_{a\in\mathcal{A}} f_1(s,a,z)^\top Q_{\text{rot}} z\right)$ do not necessarily take the same action at different states $s$ and latent $z$. Therefore, without loss of generality, when the expectations satisfy $\mathbb{E}_{\pi_1(a'|s',z)}\left[f_1(s',a',z)^\top b_1(s_f,a_f)\right] > \mathbb{E}_{\pi_2(a'|s',z)}\left[f_1(s',a',z)^\top b_1(s_f,a_f)\right]$, we can reduce the Eq. 27 to

$$\begin{aligned}
&\mathcal{T}_{\text{FB}}(f_1(s,a,z)^\top b_1(s_f,a_f)) - \mathcal{T}_{\text{FB}}(f_1(s,a,z)^\top Q_{\text{rot}}^\top Q_{\text{rot}} b_1(s_f,a_f)) \\
&= \gamma \mathbb{E}_{p(s'|s,a)}\left[\mathbb{E}_{\pi_1(a'|s',z)}\left[f_1(s',a',z)^\top b_1(s_f,a_f)\right] - \mathbb{E}_{\pi_2(a'|s',z)}\left[f_1(s',a',z)^\top b_1(s_f,a_f)\right]\right] \\
&> 0.
\end{aligned}$$

Now every element inside the $L^p$-norm on the LHS of Eq. 26 is positive, while every element inside the $L^p$-norm on the RHS is zero, which is a contradiction. Hence, we conclude that the FB Bellman operator $\mathcal{T}_{\text{FB}}$ is not a $\gamma$-contraction under the $L^p$-norm. Consequently, the Banach fixed-point theorem cannot be applied to $\mathcal{T}_{\text{FB}}$ to conclude the existence or uniqueness of a fixed point. It is unclear whether the FB algorithm (approximately) has a convergence guarantee or not. □

### C.6 Connecting One-Step FB to a Singular Value Decomposition

One intriguing interpretation of the one-step FB algorithm is that it learns the SVD of the behavioral successor measure ratio. We can make this connection precise by considering discrete MDPs with finite numbers of states and actions. Applying Lemma 1, we can compute the behavioral successor measure as

$$M^{\pi_\beta} = (1-\gamma)\left(I_{|\mathcal{S}\times\mathcal{A}|} - \gamma P^{\pi_\beta}\right)^{-1},$$

and write the behavioral successor measure ratio using notations in Appendix C.1 as

$$M^{\pi_\beta}\text{diag}(\rho)^{-1} = (1-\gamma)\left(I_{|\mathcal{S}\times\mathcal{A}|} - \gamma P^{\pi_\beta}\right)^{-1}\text{diag}(\rho)^{-1}.$$

Applying the SVD to the matrix $M^{\pi_\beta}\text{diag}(\rho)^{-1}$, we have $M^{\pi_\beta}\text{diag}(\rho)^{-1} = U_\beta \Sigma_\beta V_\beta^\top$, where $U_\beta$ and $V_\beta$ are two orthonormal matrices and $\Sigma_\beta$ is the square singular matrix. Since the behavioral successor measure ratio is fixed after fixing the behavioral policy, the one-step FB algorithm uses behavioral FB representations to fit a static target. In particular, we can set the representation dimension to $d = |\mathcal{S}\times\mathcal{A}|$ and let

$$F_\beta^\star = U_\beta \Sigma_\beta, \quad B_\beta^\star = V_\beta^\top$$

to obtain a pair of ground-truth behavioral FB representations. There are two important properties for this solution:

**Remark 2.** *The ground-truth behavioral FB representations are not unique. In particular, for a pair of ground-truth FB representations $F_\beta^\star$ and $B_\beta^\star$, and an orthonormal rotation matrix $Q_{rot} \in \mathbb{R}^{d\times d}$, $Q_{rot}F_\beta^\star$ and $Q_{rot}B_\beta^\star$ is also a pair of solution.*

**Remark 3.** *The ground-truth behavioral FB representations $F_\beta^\star = U_\beta\Sigma_\beta$ and $B_\beta^\star = V_\beta^\top$ minimizes both the TD one-step FB loss $\mathcal{L}_{\text{TD one-step FB}}$ and the orthonormalization regularization $\mathcal{L}_{ortho}$. In particular, $F_\beta^\star B_\beta^\star$ is the SVD of the behavioral successor measure ratio and $B_\beta^\star B_\beta^{\star\top} = V_\beta^\top V_\beta = I_d = I_{|\mathcal{S}\times\mathcal{A}|}$.*

### C.7 One-Step FB Enables One Step of Policy Improvement

In the same way that the FB representations are defined in Definition 2, we fit the behavioral successor measure ratio using the behavioral FB representations. Thus, the ground-truth behavioral FB representation functions $F_\beta^\star : \mathcal{S}\times\mathcal{A}\to\mathcal{Z}$ and $B_\beta^\star : \mathcal{S}\times\mathcal{A}\to\mathcal{Z}$ satisfy

$$F_\beta^\star(s,a)^\top B_\beta^\star(s_f,a_f) = \frac{M^{\pi_\beta}(s_f,a_f \mid s,a)}{\rho(s_f,a_f)}. \tag{28}$$

For a reward function $r : \mathcal{S}\times\mathcal{A}\to\mathbb{R}$, again, in the same way that the reward-specific latent variable $z_r$ is defined in Eq. 13, we define a new reward-specific latent variable $z_r^\beta$ using the behavioral backward representations as

$$z_r^\beta = \mathbb{E}_{(s_f,a_f)\sim\rho(s_f,a_f)}\left[B_\beta^\star(s_f,a_f) r(s_f,a_f)\right]. \tag{29}$$

Now, we have $z_r^\beta$ indexing the behavioral Q-value $Q_r^\beta(s,a) = F_\beta^\star(s,a)^\top z_r^\beta$ (by Eq. 28 and the relationship in Eq. 2) and the policy $\pi(a \mid s, z_r^\beta)$ performs one-step policy improvement because

$$\pi(a \mid s, z_r^\beta) = \arg\max_{a\in\mathcal{A}} F_\beta^\star(s,a)^\top z_r^\beta = \arg\max_{a\in\mathcal{A}} Q_r^\beta(s,a).$$

Importantly, one-step policy improvement does not recover the optimal policy, which is the result of multi-step policy improvement until convergence. Thus, the one-step FB algorithm loses the optimal policy adaptation property. Nevertheless, prior work [15, 29, 82, 84] has proven the success of one-step policy improvement in solving diverse RL problems. Therefore, we propose that one-step FB is a competitive method for both zero-shot adaptation (Sec. 5.3) and providing a good initialization for fine-tuning on downstream tasks (Sec. E.1).

# D Experiment Details

## D.1 Didactic Experiments for the FB Algorithm

Since the latent space $\mathcal{Z}$ contains an infinite number of latent variables, we parameterize the forward representation matrix $F_\mathcal{Z} : \mathcal{Z} \to \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times d}$ using a neural network. For the backward representations, we parameterize them as a differentiable matrix $B \in \mathbb{R}^{d\times|\mathcal{S}\times\mathcal{A}|}$. To strictly align our empirical setup with theoretical analysis in Sec. 3, we enforce a latent dimension of $d = |\mathcal{S}\times\mathcal{A}|$ and fix the rank of both representations as $|\mathcal{S}\times\mathcal{A}|$. Specifically, we consider the singular value decomposition $F_\mathcal{Z} = U_F \Sigma_F V_F^\top$ and use neural networks to predict the elements of two orthonormal matrices $U_F$ and $V_F^\top$ via the Cayley transform and singular values in $\Sigma_F$. These networks independently predict the parameters of the left orthonormal matrix ($|\mathcal{S}\times\mathcal{A}| \times |\mathcal{S}\times\mathcal{A}|$), the singular values ($|\mathcal{S}\times\mathcal{A}|$), and the right orthonormal matrix ($d \times d$). Each network consists of an MLP with $(32, 32, 32)$ units and GELU activations. The backward representation matrix $B$ is also constructed from learnable orthonormal matrices, $U_B$ and $V_B^\top$, and learnable singular values ($\Sigma_B$), as $B = U_B \Sigma_B V_B^\top$.

We train the algorithm for $10^5$ gradient steps with a batch size of 512. We set the discount factor to $\gamma = 0.9$ during training. Optimization is performed using AdamW [60] optimizer with weight decay of $10^{-4}$, $\epsilon_{\text{adamw}} = 10^{-5}$ and learning rate of $10^{-4}$. We randomly sample latent variables $z$ at each training step and use another 1000 randomly sampled latents for evaluation. Following the original FB implementation [103], we sample the latent variable $z$ from a scaled von Mises-Fisher distribution. We first sample a $d$-dimension standard Gaussian variable $x \sim \mathcal{N}(0, I_d)$ and a scalar centered Cauchy variable $u \sim \text{Cauchy}(0, 0.5)$, and then compute the latent variable as $z = \sqrt{d} u \frac{x}{\|x\|}$. We use the prior distribution $z \sim p_\mathcal{Z}(z)$ to denote this sampling procedure. The latent $z$ is preprocessed as $\tilde{z} \leftarrow \frac{z}{\sqrt{1+\|z\|_2^2/d}}$ before being passed as input to the forward representation matrix $F_\mathcal{Z}$. This transformation maps the infinite space of $\mathbb{R}^d$ to a bounded open ball of radius $\sqrt{d}$. Importantly, this mapping is a bijection that preserves differences in magnitude; therefore, latent vectors $z_r$ and $z_{\nu r+\xi}$ ($\nu > 0$) corresponding to differently scaled rewards remain distinct inputs to the neural network.

Given any marginal measure vector $\rho \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$[8], to optimize the neural networks for forward-backward representations $F_\mathcal{Z}$, $B$, we choose to use the MC FB loss $\mathcal{L}_{\text{MC FB}}$ (Eq. 4) over a batch of latent variables:

$$\mathcal{L}_{\text{MC FB}}(F_\mathcal{Z}, B) = \mathbb{E}_{z\sim p_\mathcal{Z}(z)} \left[ \left\| F_\mathcal{Z} B - M_\mathcal{Z}^\pi \text{diag}(\rho)^{-1} \right\|_F^2 \right],$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. We choose this loss because the successor measure can be computed analytically in this discrete CMP (Lemma 1): for each latent variable $z$, we have

$$M_z^\pi = (1-\gamma)\left(I_{|\mathcal{S}\times\mathcal{A}|} - \gamma P^{\pi(a|s,z)}\right)^{-1}, \quad z \sim p_\mathcal{Z}(z).$$

Thus, the MC FB loss is an analytical analogy of the TD FB loss in Eq. 6. In fact, the TD FB loss uses transition samples and target networks to approximate the MC FB loss, similar to FQE. See Sec. 3.2 for the complete discussion.

[8] We set the marginal measure to $\rho(s,a) \triangleq 1/|\mathcal{S}\times\mathcal{A}|$ in our experiments.

Instead of setting the latent-conditioned policy as the non-differentiable argmax policy: $\hat{\pi}(a \mid s, z) = \delta(a \mid \arg\max_{a \in \mathcal{A}} F(s, a, z)^\top z)$, we choose to use the Boltzmann policy

$$\hat{\pi}(a \mid s, z) = \frac{\exp(\tau_{\text{policy}} F(s, a, z)^\top z)}{\sum_{a' \in \mathcal{A}} \exp\left(\tau_{\text{policy}} F(s, a', z)^\top z\right)}, \tag{30}$$

where $\tau_{\text{policy}}$ is a temperature for the softmax function fixed to $\tau_{\text{policy}} = 5 \times 10^{-3}$ during training. Following prior practice [103], we use a temperature $\tau_{\text{policy}} = 1$ during evaluation.

We evaluate the FB algorithm by aggregating statistics over 8 random seeds using the fixed 1000 evaluation latents $D_{\text{eval}} = \{z_i\}_{i=1}^{1000}$. For each sampled latent $z$, we recover the corresponding reward vector $r \in \mathbb{R}^{\mathcal{S} \times \mathcal{A}}$ as $r = (B^{-1} z) \oslash \rho$, where $\oslash$ denotes elementwise division and the backward representation matrix $B$ is invertible (full rank). We then compute the ground-truth optimal Q-value for each reward, $Q_r^*$, by running standard value iteration until convergence. We report the following four metrics as in Fig. 3 *(Left)*:

1. **Successor measure ratio prediction error.** This metric measures the fidelity of the learned FB representations in approximating the ground-truth successor measure ratio. We define the successor measure ratio prediction error as the mean squared error (MSE) between the ratio predicted by the FB representations and the ground-truth ratio. The ground-truth ratio is computed using the reward-maximizing policy induced by the forward representations. Formally, the successor measure ratio prediction error is

$$\epsilon_{\text{SMR}} = \mathbb{E}_{z \sim D_{\text{eval}}} \left[ \left\| F_{\mathcal{Z}} B - M_{\mathcal{Z}}^{\pi} \text{diag}(\rho)^{-1} \right\|_F^2 \right].$$

2. **Optimal Q-value prediction error.** This metric measures the accuracy of the optimal Q-value predicted by the learned representation. For each latent variable $z$ with the corresponding reward vector $r$, the learned forward representation matrix $F_{\mathcal{Z}}$ predicts the optimal Q-value as $\hat{Q}_r^\star(z) = F_z z \in \mathbb{R}^{|\mathcal{S} \times \mathcal{A}|}$. On the other hand, we can compute the ground-truth optimal Q-value for reward vector $r$ using value iteration as $Q_r^\star(z) \in \mathbb{R}^{|\mathcal{S} \times \mathcal{A}|}$. The optimal Q-value prediction error is defined as the MSE between the predicted Q-value and the ground-truth Q-values:

$$\epsilon_{Q^\star} = \mathbb{E}_{z \sim D_{\text{eval}}} \left[ \left\| \hat{Q}_r^\star(z) - Q_r^\star(z) \right\|_2^2 \right].$$

3. **Forward KL divergence (optimal policy).** To evaluate the decision-making quality of the induced policy, we measure the forward KL divergence between the optimal policy derived from $Q_r^\star(z)$ and $\pi^\star(a \mid s, z)$ and the policy derived from $\hat{Q}_r(z)$ and $\hat{\pi}(a \mid s, z)$. We report the forward KL divergence averaged over all evaluation latents and all possible states:

$$\text{KL}_{\pi^\star} = \frac{1}{|\mathcal{S}|} \mathbb{E}_{z \sim D_{\text{eval}}} \left[ \sum_{s \in \mathcal{S}} D_{\text{KL}} \left( \pi^\star(\cdot \mid s, z) \parallel \hat{\pi}(\cdot \mid s, z) \right) \right]$$

   where $D_{\text{KL}}(p \parallel q) = \sum_{x \in \mathcal{X}} p(x) \log \left( \frac{p(x)}{q(x)} \right)$ is the standard KL divergence between two probability measures $p$ and $q$.

4. **Q prediction equivariance error.** This metric assesses whether the learned Q-values respect the affine equivariance property as discussed in Lemma 3 and Proposition 2. Specifically, given a latent variable $z$ with the corresponding reward vector $r$, for a positive scalar, $\nu > 0$, and an offset, $\xi \in \mathbb{R}$, the predicted Q-value should satisfy the equivariance $\hat{Q}_r(z_{\nu r + \xi}) = \hat{Q}_r(\nu z + \xi z_{\text{one}}) = \nu \hat{Q}_r(z) + \xi$, where $z_{\text{one}}$ is the latent variable corresponding to the all one reward vector. We sample $\nu$ and $\xi$ randomly and compute the following MSE:

$$\begin{aligned} \epsilon_{\text{equiv}} &= \mathbb{E}_{z \sim D_{\text{eval}}} \left[ \left\| \hat{Q}_r(\nu z + \xi z_{\text{one}}) - \left( \nu \hat{Q}_r(z) + \xi \right) \right\|_2^2 \right] \\ &= \mathbb{E}_{z \sim D_{\text{eval}}} \left[ \| F_{\nu z + \xi z_{\text{one}}} \cdot (\nu z + \xi z_{\text{one}}) - (\nu F_z z + \xi) \|_2^2 \right]. \end{aligned}$$

### D.2 Didactic Experiments for the One-Step FB Algorithm

The forward representation matrix $F_\beta \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times d}$ and the backward representation matrix $B_\beta \in \mathbb{R}^{d\times|\mathcal{S}\times\mathcal{A}|}$ are parameterized directly as differentiable matrices. We set the latent dimension of $d = |\mathcal{S}\times\mathcal{A}|$ and enforce a fixed rank of $|\mathcal{S}\times\mathcal{A}|$ on both representations. Following the construction in Appendix D.1, both matrices are factorized via SVDs, where the decompositions are formed by learnable orthonormal matrices and singular values.

We train the algorithm for $10^5$ gradient steps to fit the analytical density ratio induced by a fixed policy $\pi_\beta$. For simplicity, we set the policy $\pi_\beta$ to be a uniform policy over the entire action space: $\pi_\beta(a \mid s) \triangleq 1/|\mathcal{A}|$. Since the target density ratio $M^{\pi_\beta}\text{diag}(\rho)^{-1}$ is fixed given $\pi_\beta$, the learning one-step FB reduces to solving a supervised learning problem. Specifically, we minimize the Monte Carlo one-step FB (MC FB) loss, which resembles the MC FB loss in Eq. 4. Given the marginal measure vector $\rho \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$, the MC one-step FB loss is defined as:

$$\mathcal{L}_{\text{MC one-step FB}}(F_\beta, B_\beta) = \|F_\beta B_\beta - M^{\pi_\beta}\text{diag}(\rho)^{-1}\|_F^2, \tag{31}$$

where the fixed successor measure $M^{\pi_\beta}$ is computed as (Lemma 1)

$$M^{\pi_\beta} = (1-\gamma)\left(I_{|\mathcal{S}\times\mathcal{A}|} - \gamma P^{\pi_\beta}\right)^{-1}.$$

All optimization hyperparameters are kept identical to the FB implementation described in Appendix. D.1.

Similar to the experiments in Appendix D.1, we evaluate one-step FB by aggregating the statistics over 8 random seeds using a fixed batch of 1000 evaluation latents. For each sampled latent $z$, we also recover the corresponding reward vector $r \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ as $r = (B^{-1}z) \oslash \rho$. We then compute the Q-value of the fixed policy $Q^{\pi_\beta}$ by running standard value iteration until convergence. We report the following metrics:

- **Successor measure ratio prediction error.** The metric measures the fidelity of the learned representation in approximating the fixed successor measure ratio. It is defined as the MSE between the ratio predicted by the one-step FB representations and the ground-truth ratio.

$$\epsilon_{\text{SMR}} = \|F_\beta B_\beta - M^{\pi_\beta}\text{diag}(\rho)^{-1}\|_F^2$$

- **Q-value prediction error.** This metric measures the accuracy of the Q-value predicted by the learned representation. For each latent variable $z$ that induces the reward vector $r$, the learned forward representation matrix $F_\beta$ predicts the Q-value as $\hat{Q}_r^{\pi_\beta}(z) = F_\beta z \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$. The Q-value prediction error is defined as the MSE between the predicted Q-value and the Q-value obtained via value iteration:

$$\epsilon_{Q^{\pi_\beta}} = \mathbb{E}_{z\sim D_{\text{eval}}}\left[\left\|\hat{Q}_r^{\pi_\beta}(z) - Q_r^{\pi_\beta}(z)\right\|_2^2\right] \tag{32}$$

- **Forward KL divergence** ($\arg\max_a Q^{\pi_\beta}$)**.** We now measure the decision-making quality of the induced policy. As discussed in Sec. 4 and Appendix C.7, the latent-conditioned policy derived from the one-step FB algorithm performs one step of policy improvement over the predicted Q-value $\hat{Q}_r^{\pi_\beta}(z)$ as

$$\hat{\pi}_{\text{one-step}}(a \mid s, z) = \frac{\exp(\tau_{\text{policy}} F_\beta(s,a)^\top z)}{\sum_{a'\in\mathcal{A}}\exp(\tau_{\text{policy}} F_\beta(s,a')^\top z)} \approx \arg\max_{a\in\mathcal{A}} \hat{Q}_r^{\pi_\beta}(s,a,z).$$

This policy is trying to fit the one-step policy improvement over the ground-truth Q-value $Q_r^{\pi_\beta}(z)$. We will define the resulting policy from the one step of policy improvement over $Q_r^{\pi_\beta}(z)$ as

$$\pi_{\text{one-step}}(a \mid s, z) = \delta\left(a \mid \arg\max_{a\in\mathcal{A}} Q_r^{\pi_\beta}(s,a,z)\right).$$

To evaluate the performance of $\hat{\pi}_{\text{one-step}}$, we measure the forward KL divergence between $\pi_{\text{one-step}}$ and $\hat{\pi}_{\text{one-step}}$, averaging over all sampled latents and all possible states:

$$\text{KL}_{\pi_{\text{one-step}}} = \frac{1}{|\mathcal{S}|}\mathbb{E}_{z\sim D_{\text{eval}}}\left[\sum_{s\in\mathcal{S}} D_{\text{KL}}\left(\pi_{\text{one-step}}(\cdot \mid s, z) \parallel \hat{\pi}_{\text{one-step}}(\cdot \mid s, z)\right)\right] \tag{33}$$

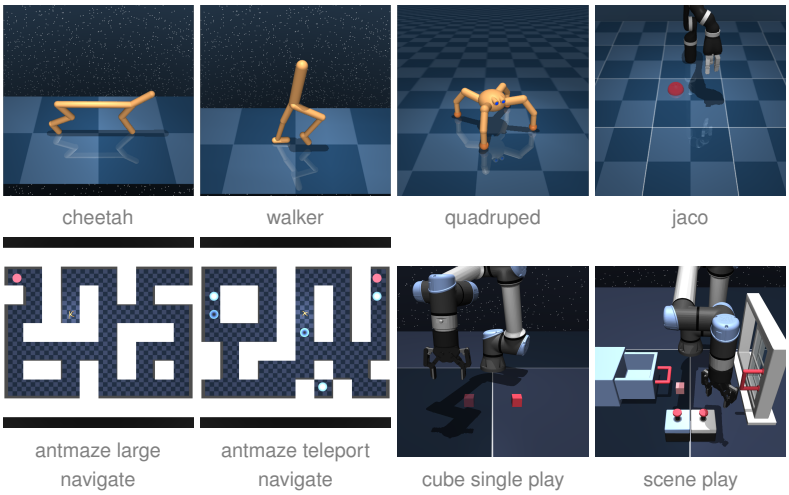



Figure 4: **Domains for evaluation.** *(Top)* ExORL domains (16 state-based tasks). *(Bottom)* OGBench domains (20 state-based tasks and 10 image-based tasks).

- **Q prediction equivariance error.** This metric assesses whether the learned Q-value respects the affine equivariance property as discussed in Lemma 3 and Proposition 2. Specifically, given a latent variable $z$ with the corresponding reward vector $r$, for a positive scalar, $\nu > 0$, and an offset, $\xi \in \mathbb{R}$, the predicted Q-value should satisfy the equivariance $\hat{Q}_r^{\pi_\beta}(z_{\nu r+\xi}) = \hat{Q}_r^{\pi_\beta}(\nu z + \xi z_{\text{one}}) = \nu \hat{Q}_r^{\pi_\beta}(z) + \xi$. We sample $\nu$ and $\xi$ randomly and compute the following MSE:

$$\begin{aligned}\epsilon_{\text{equiv}} &= \mathbb{E}_{z\sim D_{\text{eval}}}\left[\left\|\hat{Q}_r^{\pi_\beta}(\nu z + \xi z_{\text{one}}) - \left(\nu \hat{Q}_r^{\pi_\beta}(z) + \xi\right)\right\|_2^2\right] \\ &= \mathbb{E}_{z\sim D_{\text{eval}}}\left[\|F_\beta \cdot (\nu z + \xi z_{\text{one}}) - (\nu F_\beta z + \xi)\|_2^2\right].\end{aligned}$$

### D.3 Rationales of Selecting Prior Methods

We compare one-step FB against 5 prior unsupervised pre-training methods. The most relevant prior work is FB [103], which simultaneously learns a latent-conditioned policy and its occupancy measure. Other popular zero-shot methods learn state representations first and then use off-the-shelf RL algorithms (e.g., TD3 [32]) to maximize the intrinsic reward derived from those state representations. Among them, we mainly compare against two families of methods. First, some prior methods derived state representations for the successor measure using variants of the expectile regression adapted from Kostrikov et al. [50]: HILP [81] and ICVF [34]. Second, another line of established methods trains state representations via consistency along single-step or multi-step samples from the successor measure: Laplacian [107] and BYOL-$\gamma$ [54].

### D.4 Environments, Datasets, and Evaluation Protocols

We focus on the offline setting and use standard offline RL benchmarks to compare one-step FB against prior methods. We select a set of 4 state-based locomotion domains from the ExORL [109] benchmark and a set of 4 state-based robotic navigation and manipulation domains from the OGBench [80] benchmark (Fig. 4). To test whether our method is able to directly take in RGB images as inputs, we additionally include 2 image-based domains from the OGBench benchmark. Our experiments pre-train different methods for $10^6$ gradient steps and evaluate the zero-shot performance on a diverse set of tasks in each domain.

**Benchmarks.** The ExORL benchmark consists of a diverse set of locomotion tasks based on the DeepMind Control Suite [99]. Following prior work [81], we select 4 domains from the entire benchmark, each containing 4 tasks. These tasks involve controlling four robots (`cheetah`, `walker`,

quadruped, and jaco) to complete different locomotion behaviors. For each domain, the specific tasks are as follows:

- walker: flip, run, stand, and walk.
- cheetah: run, run backward, walk, and walk backward.
- quadruped: jump, run, stand, and walk.
- jaco: reach bottom left, reach bottom right, reach top left, reach top right.

For domains walker, cheetah, and quadruped, both the episode length and the maximum return are 1000. For the domain jaco, both the episode length and the maximum return are 250. As mentioned in Yarats et al. [109], tasks in walker, cheetah, and quadruped use dense reward functions, while tasks in jaco use sparse reward functions. Detailed descriptions of the benchmark can be found in Yarats et al. [109]. Thus, zero-shot adaptation on jaco is more challenging than on other domains.

The OGBench benchmark consists of a diverse set of robotic navigation and manipulation tasks. These tasks are built on top of the MuJoCo simulator [102] and are designed for goal-conditioned control. We select 4 state-based domains and 3 image-based domains, each containing 5 tasks. The goal of these tasks is to either control an Ant to navigate in deterministic or stochastic mazes (antmaze large navigate, antmaze teleport navigate, and visual antmaze medium navigate) or control a robot arm to rearrange various objects (cube single play, scene play, visual cube single play, and visual scene play). For each state-based domain, the specific tasks are:

- antmaze large navigate: task 1 (bottom left to top right), task 2 (center to top left), task 3 (center to bottom right), task 4 (bottom right to center), and task 5 (bottom left to center).
- antmaze teleport navigate: task 1 (bottom right to top left), task 2 (bottom left to top right), task 3 (center to top right), task 4 (top left to top right), and task 5 (center to top left).
- cube single play: task 1 (pick and place to left), task 2 (pick and place to front), task 3 (pick and place to back), task 4 (pick and place diagonally), and task 5 (pick and place off-diagonally).
- scene play: task 1 (open drawer and window), task 2 (close and lock drawer and window), task 3 (open drawer, close window, and pick and place cube to right), task 4 (put cube in drawer), and task 5 (fetch cube from drawer and close window).

For each image-based domain, the specific tasks are the same as the state-based variant, except visual antmaze medium navigate. The visual antmaze medium navigate domain uses local third-person image observations as input to algorithms and includes ground and wall colors for agents to infer their locations. This domain contains five tasks: task 1 (bottom left to top right), task 2 (top left to bottom right), task 3 (turn around central corner), task 4 (top right to top left), and task 5 (bottom right to bottom left). All visual observations are $64 \times 64 \times 3$ RGB images. These tasks are challenging because the agent must reason directly from pixels. For domains involving navigation tasks antmaze large navigate and antmaze teleport navigate, the maximum episode length is 1000. For tasks in cube single play, the maximum episode length is 200. For tasks in scene play, the maximum episode length is 750. These domains are challenging because they all use sparse goal-conditioned reward functions. For other details of the benchmark, please refer to Park et al. [80].

**Datasets.** On the ExORL benchmark, following the prior work [104, 81, 47, 115], we use $5 \times 10^6$ transitions collected by an exploration method (RND [17]) for unsupervised pre-training, and another $10^5$ transitions collected by the same exploratory policy for zero-shot inference (predicting $z_r$). The zero-shot adaptation datasets will be labeled with task-specific dense rewards, except in jaco, where the reward signals are sparse. See Yarats et al. [109] for details of data collection.

On the OGBench benchmark, following the prior work [115, 5], for both state-based and image-based tasks, we use $10^6$ transitions collected by a non-Markovian expert policy with temporally correlated noise (the play datasets) for unsupervised pre-training, and another $10^5$ transitions collected by

the same noisy expert policy for zero-shot inference. Unlike the ExORL benchmark, the zero-shot adaptation datasets will be labeled with *semi-sparse* rewards [83]. See Park et al. [80] for details of data collection.

**Evaluation Protocols.** We compare the performance of one-step FB against the 5 baselines (Sec. D.3) by pre-training each method for $10^6$ gradient steps on different domains ($5 \times 10^5$ gradient steps for image-based domains). We simultaneously perform zero-shot inference to measure the performance of each method. During zero-shot adaptation, we relabel the $10^5$ transitions (Appendix D.4) with task-specific rewards and use them to infer the latent variable $z$ among different methods. Note that for OGBench domains, we use the semi-sparse reward instead of a success indicator for zero-shot inference. After inferring the latent variable $z_r$, we fix it inside the latent-conditioned policy $\pi(a \mid s, z_r)$ and use the policy to do evaluation. On domains from the ExORL benchmark, we measure the undiscounted cumulative return averaged over 50 episodes. On domains from the OGBench benchmark, we measure the success rate average over 50 episodes. Following prior practice [83, 98], we do *not* report the best performance during pre-training and instead report the evaluation results averaged over $8 \times 10^5$, $9 \times 10^5$, and $10^6$ gradient steps for state-based domains. For image-based tasks, we report the evaluation results averaged over $4 \times 10^5$, $4.5 \times 10^5$, and $5 \times 10^5$ gradient steps. Following prior work [83], we report means and standard deviations over 8 random seeds for state-based domains (4 seeds for image-based domains).

For offline-to-online fine-tuning, we only compare one-step FB to prior methods on state-based tasks. We first use the zero-shot policy inferred by different methods as the initialization and then fine-tune the policy for $10^6$ environment steps (environment step = gradient step) using TD3 [32]. Note that we do not retain offline data in the online replay buffer because the offline data lacks reward signals. Again, we measure the undiscounted cumulative return for tasks from the ExORL benchmark and measure the success rate for tasks from the OGbench benchmark. We evaluate the performance of the fine-tuned policy every $10^5$ environment steps. For completeness, we show the full learning curves aggregated over 8 random seeds.

### D.5 Implementations and Hyperparameters

We compare one-step FB against 5 baselines, measuring the performance of undiscounted cumulative returns and success rates on downstream tasks. We implement one-step FB and all baselines using JAX [14], adapting the OGBench [80] codebase. Our open-source implementations can be found at `https://github.com/chongyi-zheng/onestep-fb`. All experiments for state-based domains ran on a single A6000 GPU for up to 6 hours, and all experiments for image-based domains ran on the same type of GPU for up to 16 hours.

Following prior work [101, 104], we apply two common practices that improve the overall performance of every method. First, the prior measure over latent variables $p_{\mathcal{Z}}(z)$ is set to a scaled von Mises-Fisher distribution: we first sample from the standard Gaussian distribution of $d$ dimensions $x \sim \mathcal{N}(0, I_d)$ and then normalize and rescale the sample to obtain a latent variable $z = \sqrt{d}x/\|x\|_2$. Second, when sampling latent variables for training, we include both latents from the prior distribution $p_{\mathcal{Z}}(z)$ and latents constructed from the current representations. Specifically, for FB and one-step FB, we use the normalized and rescaled variants of backward representations to construct latents. For all other baselines, we use the normalized and rescaled variants of state representations to construct latents. These constructed latents are mixed with latents sampled from the prior distribution with 0.5 probability to form the final latents for pre-training. In addition to these common practices, each method adopts specific implementation details, which we describe below.

**One-step FB.** The one-step FB consists of three main components for unsupervised pre-training: the forward representation $F_\beta$, the backward representation $B_\beta$, and the latent-conditioned policy $\pi(a \mid s, z)$. We model all of them as multilayer perceptrons (MLPs) and use different architectures for the ExORL domains and the OGBench domains, respectively (See Table 3). We learn the forward-backward representations using the TD one-step FB loss and the orthonormalization regularization (Eq. 9). Following prior work [80], the policy network outputs the mean and the standard deviation of a Gaussian distribution with a `tanh` transformation to predict actions. We learn this Gaussian policy using the policy loss in Eq. 10. Following prior work [101], before inferring the latent variable $z_r^\beta$ using zero-shot transitions (Eq. 29), we apply softmax weights based on the rewards of these

Table 3: Common hyperparameters for one-step FB and prior methods.

| Hyperparameter | Value |
|---|---|
| optimizer | Adam [48] |
| batch size | 1024 on state-based domains, 256 on image-based domains |
| learning rate | $1 \times 10^{-4}$ |
| actor MLP hidden layer sizes | $(1024, 1024, 1024)$ on ExORL domains<br>$(512, 512, 512, 512)$ on OGBench domains |
| value MLP hidden layer sizes | $(1024, 1024, 1024)$ on ExORL domains<br>$(512, 512, 512, 512)$ on OGBench domains |
| representation MLP hidden layer sizes | Laplacian, BYOL-$\gamma$, ICVF, and HILP: $(256, 256, 256)$ on ExORL domains<br>FB and one-step FB: $(1024, 1024, 1024)$ for forward representations on ExORL domains,<br>$(512, 512)$ for backward representations on ExORL domains<br>All methods: $(512, 512, 512, 512)$ on OGBench domains |
| MLP layer normalization | No |
| MLP activation function | ReLU [72] on ExORL domains<br>GELU [40] on OGBench domains |
| discount factor $\gamma$ | 0.98 on ExORL domains, 0.99 on OGBench domains |
| target networks update coefficient | $1 \times 10^{-2}$ on ExORL domains, $5 \times 10^{-3}$ on OGBench domains |
| representation dimension $d$ | 50 on ExORL domains, 128 on OGBench domains |
| latent mixing probability | 0.5 |
| actor tanh transformation | Yes |
| TD3 action noise distribution | $\texttt{clip}(\mathcal{N}(0, 0.2^2), -0.2, 0.2)$ |
| image encoder | small IMPALA encoder [24, 83] |
| image augmentation method | random cropping |
| image augmentation probability | 0.5 |
| image frame stack | 3 |

transitions $\{(s_i, a_i, r_i)\}_{i=1}^N$, resulting in the following new rewards:

$$\tilde{r}(s_i, a_i) = w_i \cdot r(s_i, a_i), w_i = \frac{\exp\left(\tau_{\text{reward}} \cdot r(s_i, a_i)\right)}{\sum_{j=1}^N \exp\left(\tau_{\text{reward}} \cdot r(s_j, a_j)\right)}, \tag{34}$$

where $\tau_{\text{reward}}$ is the temperature. For representation dimension $d$, we set it to 50 for ExORL domains and set it to 128 for OGBench domains. In our initial experiments, we found that one-step FB's performance is sensitive to the behavioral-cloning regularization coefficient $\lambda_{\text{BC}}$, the orthonormalization regularization coefficient $\lambda_{\text{ortho}}$, and the reward weighting temperature $\tau_{\text{reward}}$. We perform hyperparameter sweeps over $\lambda_{\text{BC}} \in \{0, 0.03, 0.3, 3, 30\}$, $\lambda_{\text{ortho}} \in \{0, 0.03, 0.3, 1\}$, and $\tau_{\text{reward}} \in \{3, 10, 30\}$ to select the best values for each domain. We summarize the hyperparameters for one-step FB in Table 3 and Table 4.

**FB [103].** Our FB implementation adapts the implementation from Tirinzoni et al. [101] and is similar to the one-step FB implementation. There are two main differences between the FB implementation and the one-step FB implementation. First, the forward representation network in FB takes in the latent variable $z$ as input. Second, when computing the forward-backward representation loss (Eq. 6), the next action $a'$ is sampled from the latent-conditioned policy $\pi(a' \mid s', z)$. During the zero-shot adaptation, similar to one-step FB, we compute the latent variable $z_r$ for a downstream task as in Eq. 13. We use the same representation dimension as in the one-step FB implementation for consistency. We also perform hyperparameter sweeps over $\lambda_{\text{BC}} \in \{0, 0.03, 0.3, 3, 30\}$, $\lambda_{\text{ortho}} \in \{0, 0.03, 0.3, 1\}$, and $\tau_{\text{reward}} \in \{3, 10, 30\}$ to select the best values for each domain. See Table 3 and Table 4 for details of hyperparameters.

Besides the FB algorithm, we also compare one-step FB against unsupervised pre-training methods for RL that first learn state representations $\phi : \mathcal{S} \to \mathcal{Z}$ and then use off-the-shelf RL algorithms to maximize intrinsic rewards derived from the state representations $r(s, z) = \phi(s)^\top z$. During the zero-shot adaptation, all methods (except one-step FB and FB) find the appropriate latent variable $z_r$ by solving a simple linear regression problem [7, 81]:

$$z_r = \arg\min_{z \in \mathcal{Z}} \mathbb{E}_{\rho(s,a)} \left[ \left( r(s, a) - \phi(s)^\top z \right)^2 \right],$$

where $\rho \in \Delta(\mathcal{S} \times \mathcal{A})$ is the marginal measure over states and actions as in one-step FB. For HILP and ICVF, both methods learn state representations using an expectile regression loss adapted from IQL [50]. They differ in how they decompose the successor measure. For BYOL-$\gamma$ and Laplacian,

Table 4: **Domain-specific hyperparameters for one-step FB and prior methods.** Following prior work Park et al. [83], we tune these hyperparameters for each domain from ExORL and OGBench benchmarks. "-" indicates the hyperparameters do not exist. The complete description of each hyperparameter can be found in Appendix D.5.

| | Laplacian | | BYOL-$\gamma$ | | ICVF | | HILP | | FB | | | One-Step FB | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Domain | $\lambda_{\text{BC}}$ | $\lambda_{\text{ortho}}$ | $\lambda_{\text{BC}}$ | $\lambda_{\text{ortho}}$ | $\lambda_{\text{BC}}$ | $\mu$ | $\lambda_{\text{BC}}$ | $\mu$ | $\lambda_{\text{BC}}$ | $\lambda_{\text{ortho}}$ | $\tau_{\text{reward}}$ | $\lambda_{\text{BC}}$ | $\lambda_{\text{ortho}}$ | $\tau_{\text{reward}}$ |
| `walker` | 30 | 1 | 10 | 0.01 | 0.03 | 0.5 | 0.3 | 0.9 | 0 | 0.03 | 10 | 0 | 0.1 | 10 |
| `cheetah` | 10 | 1 | 3 | 0.01 | 0.3 | 0.5 | 3 | 0.5 | 0 | 0.3 | 10 | 0 | 1 | 3 |
| `quadruped` | 10 | 1 | 3 | 0.01 | 0.03 | 0.5 | 3 | 0.9 | 0 | 1 | 10 | 0 | 0.03 | 3 |
| `jaco` | 30 | 1 | 10 | 1 | 0.03 | 0.5 | 0.3 | 0.9 | 0 | 0.3 | 10 | 0 | 0.03 | 3 |
| `antmaze large navigate` | 1 | 0.1 | 3 | 0 | 3 | 0.5 | 1 | 0.5 | 0.03 | 0 | 10 | 0.03 | 0 | 10 |
| `antmaze teleport navigate` | 30 | 0 | 1 | 0 | 0.3 | 0.5 | 1 | 0.5 | 0.03 | 0 | 3 | 0.1 | 0 | 10 |
| `cube single play` | 1 | 0 | 30 | 0 | 30 | 0.5 | 1 | 0.5 | 0.3 | 0 | 10 | 0.3 | 0.3 | 300 |
| `scene play` | 1 | 0 | 3 | 0 | 0.3 | 0.5 | 1 | 0.9 | 0.3 | 0 | 3 | 0.3 | 0 | 300 |
| `visual cube single play` | - | - | 30 | 0 | - | - | 1 | 0.5 | 1 | 0 | 300 | 1 | 0 | 300 |
| `visual scene play` | - | - | 3 | 0 | - | - | 3 | 0.5 | 0.3 | 0 | 300 | 0.3 | 0 | 10 |

both methods learn state representations via consistency over successor measures (latent predictive loss). They differ in that the consistency is either along single-step or multi-step samples from the successor measure. After learning state representations, we use the same TD3 + BC [31] algorithm to maximize the intrinsic reward for all these baselines. The TD3 + BC implementation uses a target actor to select actions in the critic loss. We also add a clipped Gaussian noise $\texttt{clip}(\mathcal{N}(0, 0.2^2), -0.2, 0.2)$ to introduce some noise into these actions. Similar to Eq. 10, the actor loss maximizes Q predicted by the critic while being regularized to output the behavioral actions via a behavioral-cloning regularization. Below, we describe the details of each method.

**HILP [81].** The HILP implementation is adapted from the official implementation [81]. The motivation of representation learning in HILP is to encode the temporal (goal-conditioned) distance between pairs of states into the Euclidean distance in a $d$-dimensional representation space. To achieve this goal, the HILP learns state representations $\phi(s)$ using the following expectile loss:

$$\mathcal{L}_{\text{HILP}}(\phi) = \mathbb{E}_{p^{\pi_\beta}(s,s'),p_{\mathcal{G}}(s_f|s)} \left[ L_2^\mu \left( -\mathbb{1}(s \neq s_f) + \gamma \|\bar{\phi}(s_f) - \bar{\phi}(s')\|_2 + \|\phi(s_f) - \phi(s)\|_2 \right) \right],$$

where $L_2^\mu(x) = |\mu - \mathbb{1}(x < 0)|x^2$ is the expectile loss with $\mu \in [0.5, 1)$, $\bar{\phi}$ is the target state representation. The future state $s_f$ is sampled from either the behavioral successor measure with probability $0.625$ or a random uniform measure with probability $0.375$, which we denote as the goal measure $p_{\mathcal{G}}(s_f \mid s)$. For the representation dimension, we set $d = 50$ for ExORL domains and set $d = 128$ for OGBench domains. We sweep over the behavioral-cloning regularization coefficient $\lambda_{\text{BC}} \in \{0.03, 0.3, 3\}$ and the expectile $\mu \in \{0.5, 0.9\}$ for different domains. See Table 3 and Table 4 for details of hyperparameters.

**ICVF [34].** The ICVF learns state representations similar to HILP, but uses a different decomposition of the intention-conditioned successor measure. Specifically, ICVF uses an intention-conditioned value $V : \mathcal{S} \times \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ to model the successor measure of visiting the future state $s_f$ starting from the current state $s$ by following the intention goal $g$. Both the future state $s_f$ and the intention goal $g$ are sampled from the goal measure similar to HILP. The intention-conditioned value decomposes as $V(s, s_f, g) = \phi(s)^\top T_{\text{ICVF}}(g)\psi(s_f)$, where $T_{\text{ICVF}} : \mathcal{S} \to \mathcal{Z} \times \mathcal{Z}$ predicts a latent transition matrix and $\psi : \mathcal{S} \to \mathcal{Z}$ is the future state representations. ICVF learns the intention-conditioned value using the following variant of the expectile loss:

$$\mathcal{L}_{\text{ICVF}}(\phi, T_{\text{ICVF}}, \psi) = \mathbb{E}_{p^{\pi_\beta}(s,s'),p_{\mathcal{G}}(s_f|s),p_{\mathcal{G}}(g|s)} \left[ |\mu - \mathbb{1}\left(A(s,s',g) < 0\right)| \left(V(s,s_f,g) - \mathbb{1}(s = s_f) - \gamma V(s',s_f,g)\right)^2 \right],$$

where $\mu \in [0.5, 1)$ and the advantage $A(s, s', g)$ is defined as

$$A(s, s', g) = \mathbb{1}(s = g) + \gamma V(s', g, g) - V(s, g, g).$$

For the representation dimension, we use the same values as in HILP for consistency. We sweep over the behavioral-cloning regularization coefficient $\lambda_{\text{BC}} \in \{0.03, 0.3, 3\}$ and the expectile $\mu \in \{0.5, 0.7, 0.9\}$ for different domains. See Table 3 and Table 4 for details of hyperparameters.

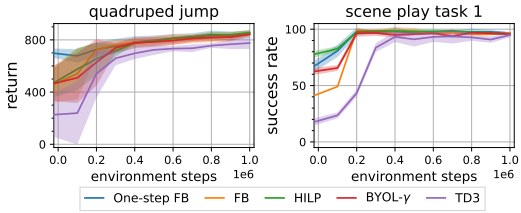



Figure 5: **Fine-tuning pre-trained agents on downstream tasks.** After offline pre-training, we conduct online fine-tuning on various methods using the same off-the-shelf RL algorithm (TD3). One-step FB continues to provide higher sample efficiency (+40% on average) during fine-tuning, as compared with the original FB method.

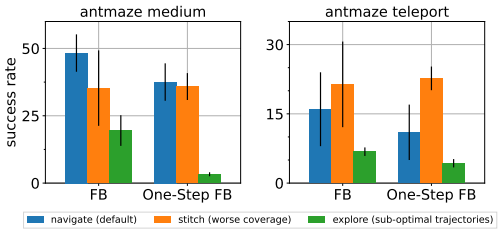



Figure 6: **The effects of the dataset quality on FB and one-step FB.** Using `stitch` dataset has minor effects on both FB and one-step FB, and sometimes even improves the performance, while using `explore` datasets reduces performance for both methods.

**BYOL-$\gamma$ [54].** BYOL-$\gamma$ learns state representations via consistency (or a latent predictive loss) over the current state and a future state sampled from the behavioral occupancy measure. Specifically, BYOL-$\gamma$ minimizes the mean squared error between state representations of the current state $s$ and the future state $s_f$ with a latent transition function $T_{\text{BYOL-}\gamma} : \mathcal{Z} \times \mathcal{A} \to \mathcal{Z}$,

$$\mathcal{L}_{\text{BYOL-}\gamma}(\phi, T_{\text{BYOL-}\gamma}) = \mathbb{E}_{p^{\pi_\beta}(s,a), M^{\pi_\beta}(s_f|s,a)} \left[ \| T_{\text{BYOL-}\gamma}(\phi(s), a) - \bar{\phi}(s_f) \|_2^2 \right],$$

where $\bar{\phi}$ is the target state representation, and $M^{\pi_\beta}(s_f \mid s, a)$ is the behavioral successor measure following a geometric distribution. We also include an orthonormalization regularization loss as in Eq. 8 to regularize the covariance of state representations. For the representation dimension, we follow the same values as in HILP. In our experiments, we sweep over the behavioral-cloning regularization coefficient $\lambda_{\text{BC}} \in \{0.03, 0.3, 3, 30\}$ and the orthonormalization regularization coefficient $\lambda_{\text{ortho}} \in \{0.01, 0.1, 1.0\}$ to find the best values for different domains. We include complete hyperparameters in Table 3 and Table 4.

**Laplacian [107].** Unlike BYOL-$\gamma$, Laplacian learns state representations via consistency over the current state and the immediate next state. Specifically, the Laplacian minimizes the mean squared error between state representations of the current state $s$ and the next state $s'$:

$$\mathcal{L}_{\text{Laplacian}}(\phi) = \mathbb{E}_{p^{\pi_\beta}(s,s')} \left[ \| \phi(s) - \phi(s') \|_2^2 \right].$$

Note that we do not use a latent transition function or a target state representation in this loss. Although state representations collapsed to a constant admitting a minimizer of this loss function, we do not observe this behavior in our experiments. Again, we include an orthonormalization regularization loss as in Eq. 8 to regularize the covariance of state representations towards the identity matrix. For the representation dimension, we use the same values as in HILP. In our experiments, we sweep over the behavioral-cloning regularization coefficient $\lambda_{\text{BC}} \in \{0.3, 1, 3, 10, 30\}$ and the orthonormalization regularization coefficient $\lambda_{\text{ortho}} \in \{0.01, 0.1, 1.0\}$ to find the best values for different domains. We include complete hyperparameters in Table 3 and Table 4.

# E Additional Experiments

## E.1 Does One-Step FB Enable Efficient Fine-Tuning?

While our prior experiments in Sec. 5.3 focus on zero-shot performance, we also want to study whether one-step FB enables efficient online fine-tuning. To test this hypothesis, we conduct offline-to-online experiments on one task in the ExORL benchmarks (`quadruped jump`) and another task taken from the OGBench benchmarks (`scene task 1`). Following the evaluation protocols in Appendix D.4, we first use $10^5$ zero-shot transitions to derive the latent variable $z_r$ for the policy $\pi(a \mid s, z_r)$, and then use it to initialize a TD3 [32] agent for fine-tuning. We selectively compare one-step FB against 3 baselines in our offline experiments: BYOL-$\gamma$, HILP, and FB, and also include the performance of a TD3 agent trained from scratch for reference.

As shown in Fig. 5, the policy derived from one-step FB achieves strong fine-tuning performance compared to alternative unsupervised pre-training, with 40% higher sample efficiency on average.

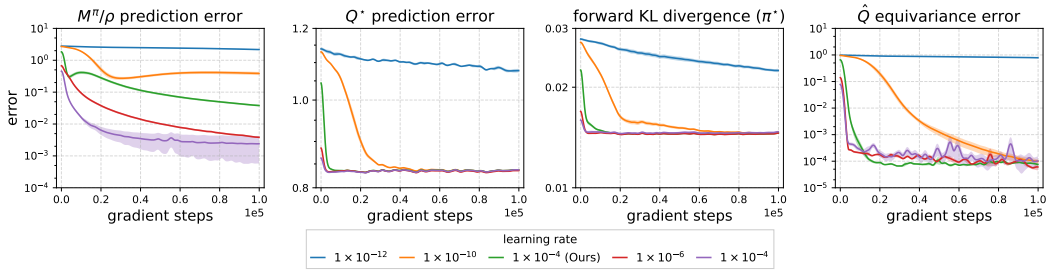



Figure 7: **Learning rate ablations for FB on the three-state CMP.** We conduct ablations to study the effect of learning rate on FB. Using learning rates other than $10^{-4}$ results in higher equivariance error of Q-value prediction ($\epsilon_{\text{equiv}} > 10^{-4}$). Thus, the failure mode of FB is *not* explained by the choice of learning rate.

Compared with the TD3 variant trained from scratch, the sample efficiency of fine-tuning the policy derived from one-step FB is $+2.25\times$ higher, suggesting the importance of the unsupervised pre-training phase. We also observe that the fine-tuned policies reach the asymptotic performance of TD3 at the end of training. Interestingly, we do *not* observe a degradation in performance at the beginning of fine-tuning; an observation that has been identified in prior offline-to-online methods [73]. In particular, we do not retain the pre-training data when performing online fine-tuning [116], helping to explain this observation. Taken together, one-step FB is a simpler method that provides benefits for both offline unsupervised pre-training and online fine-tuning.

### E.2 The Effects of the Dataset Quality on FB and One-Step FB

Since one-step FB performs one step of policy improvement over the behavioral datasets, it is important to examine the effects of dataset quality on our algorithm. We hypothesize that our method can be effective when *(1)* the dataset has good coverage of the state and action spaces, and *(2)* the dataset contains nearly optimal trajectories.

To test these hypotheses, we conduct ablations on two types of OGBench datasets: `stitch` (worse coverage on low-reward regions) and `explore` (highly sub-optimal trajectories). Specifically, we select two domains from the OGBench benchmarks, `antmaze medium` and `antmaze teleport`, and compare the zero-shot performances of both FB and one-step FB on three different types of datasets: `navigate` (default), `stitch`, and `explore`.

Results in Fig. 6 suggest that using the `stitch` dataset has minor effects on both FB and one-step FB, and sometimes even improves the performance, while using `explore` datasets reduces performance for both methods. We conjecture that worse coverage may not be a main issue when the dataset contains high-reward transitions. In contrast, highly sub-optimal trajectories introduce noise when inferring the latent variable $z$ during policy adaptation, which significantly lowers the zero-shot performance.

### E.3 Confounding Effects in the Didactic Experiments

In this section, we provide a more detailed discussion of potential confounding effects in our didactic experiments. The aim is to clarify the observation that the practical FB algorithm fails to converge, while our one-step FB algorithm converges to its stated fixed-point.

**Learning rate.** *Does the learning rate affect the convergence of FB?* We ablate over different learning rates within $\{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$, choosing a wider range than the learning rate $10^{-4}$ used in Sec. 5.1 and Sec. 5.2. As seen in Fig. 7, using learning rates either larger than or smaller than $10^{-4}$ results in a higher $\hat{Q}$ equivariance error ($\epsilon_{\text{equiv}} > 10^{-4}$), indicating that the original FB algorithm still does *not* converge to the ground-truth fixed point. These observations are consistent with our conclusions in Sec. 5.1.

**Policy temperature.** *Does the policy temperature $\tau_{policy}$ affect the convergence of FB?* As mentioned in Appendix D.1, we use a softmax policy (Eq. 30) with temperature $\tau_{\text{policy}}$ to approximate the greedy policy with respect to the inner product $F(s, a, z)^\top z$. One confounding factor is the choice of the

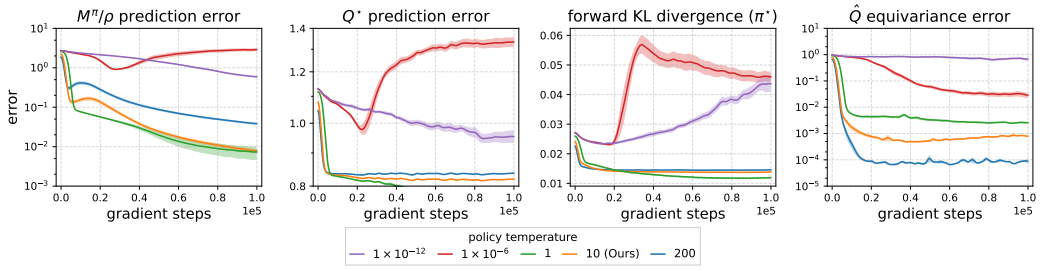



Figure 8: **Policy temperature ablations for FB on the three-state CMP.** We study the effect of policy temperature $\tau_{\text{policy}}$ on the convergence of FB: a decreasing $\tau_{\text{policy}}$ results in an increasing Q prediction equivariance error $\epsilon_{\text{equiv}}$, suggesting that FB still fails to converge to the ground-truth fixed point.

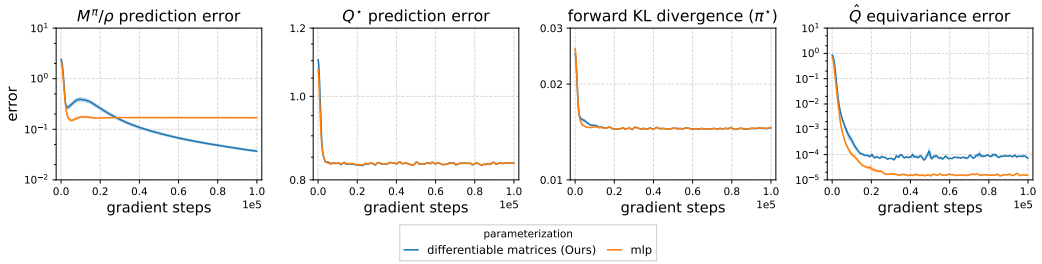



Figure 10: **Representation parameterization ablations for FB on the three-state CMP.** We study the effect of representation parameterization on the convergence of FB: using differentiable matrices yields lower error in successor measure ratio predictions, while using MLP parameterizations results in lower equivalence errors in the Q-value prediction. However, the similar failure mode for FB's convergence persists.

temperature $\tau_{\text{policy}}$. We conduct ablation experiments studying the effects of the policy temperature $\tau_{\text{policy}}$ on the convergence of FB using the same three-state CMP as in Sec. 5.1. Specifically, we choose to sweep over $\tau_{\text{policy}} \in \{1, 10^{-6}, 10^{-3}, 1, 10, 200\}$, showing results in Fig. 8. We observe that a decreasing $\tau_{\text{policy}}$ results in an increasing Q prediction equivariance error $\epsilon_{\text{equiv}}$, suggesting that $\tau_{\text{softmax}}$ is an important hyperparameter balancing learnability and convergent accuracy. However, FB still fails to converge to a ground-truth fixed point.

**Representation parameterization.** *Does a different parameterization of the FB representations affect its convergence?* As mentioned in Appendix D.1, we used differentiable matrices $F_z \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times d}$ and $B \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times d}$ to represent the FB representations for a latent variable $z$. This parameterization simplifies the optimization procedure. To study the effects of this parameterization, we conduct ablation experiments using the same three-state CMP as in Sec. 5.1. We choose to compare the differentiable matrix parameterization against a monolithic feed-forward neural network (MLP) parameterization. Results in Fig. 10 show that using differentiable matrices yields lower error in successor measure ratio predictions, while using MLP parameterizations results in lower equivalence errors in the Q-value prediction. However, the similar failure mode for FB's convergence persists.

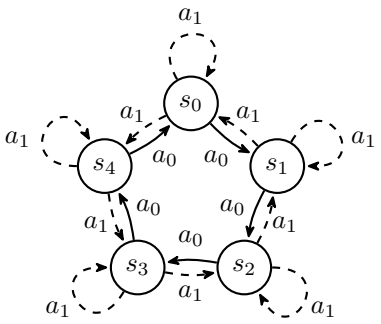



Figure 9: **The five-state circular CMP.** Agents start from state $s_0$ and take action $a_i$ $(i = 0, 1)$. At every state $s_i$, choosing $a_0$ convergence transits to the next state $s_{(i+1) \mod 5}$, forming circular transitions. At every state $s_i$, choosing action $a_1$ transits to state $s_{(i-1) \mod 5}$ with a probability of 0.7 and stays in the same state with a probability of 0.3, forming the stochastic transitions. Appendix E.3 uses this simple CMP to study the convergence of the FB and the one-step FB algorithms.

**Representation learning objective.** *Does directly optimizing the TD LSIF loss (Eq. 6) with a target network help FB converge?* Since our didactic experiments use a CMP with a discrete state and action

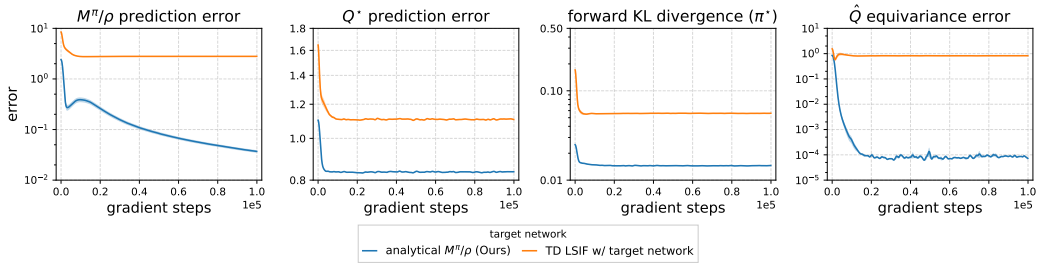



Figure 11: **Representation learning objective ablations for FB on the three-state CMP.** We conduct ablations to compare two representation learning objectives in our didactic experiments: *(1)* analytically compute the successor measure ratio and conduct bilinear decomposition into FB representations, and *(2)* learn the FB representations using the TD LSIF loss (Eq. 6) directly. While the former learning objective consistently achieved lower errors on different metrics than the latter objective. The final learned FB representations still failed to converge to the ground-truth FB representations.

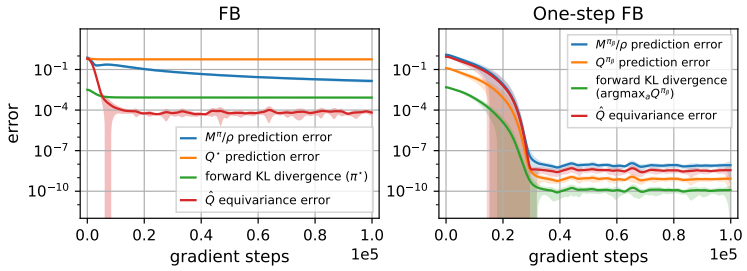



Figure 12: **Learning FB representations in the five-state circular CMP.** *(Left)* After training for $10^5$ gradient steps, FB fails to converge to a pair of ground-truth FB representations. *(Right)* Given a fixed policy, one-step FB exactly fits the ground-truth one-step FB representations within $4 \times 10^4$ gradient steps. These observations are consistent with our analysis on the three-state CMP (Sec. 5.1 and Sec. 5.2).

space, we choose to compute the successor measure ratio $M_{\mathcal{Z}}^{\pi}\mathrm{diag}(\rho)^{-1}$ analytically, and fit the FB representations $F_{\mathcal{Z}}$ and $B$ using the mean squared error (See Appendix D.1). However, the practical FB algorithm optimizes the TD LSIF loss (Eq. 6) directly without computing the successor measure ratio (impossible to compute for continuous CMPs). We conduct ablation experiments comparing the effects of using these two objectives for learning FB representations on FB's convergence. Results on the same three-state CMP (Fig. 11) show that while analytically computing the successor measure ratio consistently achieved lower errors on different metrics than the TD LSIF loss. The final learned FB representations still failed to converge to the ground-truth FB representations.

**Different discrete CMPs.** *Is the three-state CMP a special case where FB fails to converge?* To rule out the confounding factors originating from the choice of CMPs, we conduct additional didactic experiments on a new CMP, similar to Sec. 5.1 and Sec. 5.2. Specifically, we construct a five-state circular CMP (Fig. 9), where agents start from state $s_0$ and take action $a_i$ $(i = 0, 1)$. At every state $s_i$, choosing $a_0$ convergence transits to the next state $s_{(i+1) \mod 5}$, forming circular transitions. At every state $s_i$, choosing action $a_1$ transits to state $s_{(i-1) \mod 5}$ with a probability of $0.7$ and stays in the same state with a probability of $0.3$, forming the stochastic transitions.

As shown in Fig. 12, we track the important statistics for both FB and one-step FB, similar to Sec. 5.1 and Sec. 5.2. Importantly, in this new five-state circular CMP, we observe convergences similar to those in the three-state CMP for both methods. These results also highlight that FB fails to converge to a pair of ground-truth FB representations, while one-step FB exactly fits the ground-truth one-step FB representations within $4 \times 10^4$ gradient steps. Taken together, our conclusions are consistent across different didactic CMPs.

### E.4 Key Components of One-Step FB

In this section, we conduct ablation experiments studying key components of one-step FB. We choose two ExORL domains `walker` and `cheetah`, and two OGBench domains `antmaze large navigate` and `scene play` to conduct experiments. As mentioned in Sec. D.5, there are four key

hyperparameters of one-step FB: *(1)* the behavioral-cloning regularization coefficient $\lambda_{\text{BC}}$, *(2)* the orthonormalization regularization coefficient $\lambda_{\text{ortho}}$, *(3)* the reward weighting temperature $\tau_{\text{reward}}$, and *(4)* the representation dimension $d$. Following the same evaluation protocol as in Appendix D, we compute means and standard deviations over 8 random seeds for each domain.

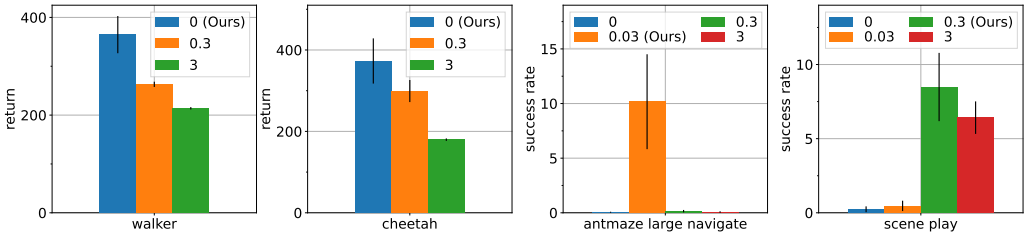
We first study the effects of the behavioral-cloning regularization coefficient $\lambda_{\text{BC}}$, ablating over different values of $\lambda_{\text{BC}}$ within $\{0, 0.03, 0.3, 3, 30\}$. We selectively show the comparisons between several $\lambda_{\text{BC}}$ values in Fig. 13a. These results suggest that removing the behavioral-cloning regularization $\lambda_{\text{BC}} = 0$ is important on ExORL domains, where our choices achieved $+1.6\times$ improvement on average. This observation is consistent with findings from prior work [81], where they also exclude the behavioral-cloning regularization on ExORL tasks. In contrast, on OGBench domains, selecting a non-zero $\lambda_{\text{BC}}$ boosts the performance.

Next, to better understand the role of orthonormalizing the backward representations, we ablate over different values of the orthonormalization coefficient $\lambda_{\text{ortho}} \in \{0.01, 0.1, 0.0, 1.0, 10.0\}$ and compare the zero-shot performance of one-step FB variants. We present the results with selective values of $\lambda_{\text{ortho}}$ in Fig. 13b. Overall, the performance of one-step FB is sensitive to the choice of $\lambda_{\text{ortho}}$ on both ExORL and OGBench domains. We observe that setting $\lambda_{\text{ortho}} < 0.1$ results in lower performance on ExORL domains used for ablation experiments. In contrast, using a very large $\lambda_{\text{ortho}}$ has negative effects on one-step FB for OGBench domains. Additionally, we observe that simply removing the orthonormalization regularization ($\lambda_{\text{ortho}} = 0$) can boost the success rate by at most $+2\times$ on OGBench domains. This indicates that using an appropriate value of $\lambda_{\text{ortho}}$ is key to the performance of one-step FB.

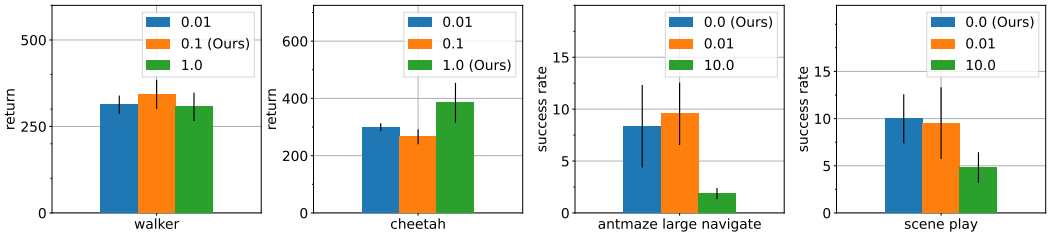
We also study the effect of our reward weighting strategy in Eq. 34 by ablating over different temperatures, $\tau_{\text{reward}} \in \{3, 10, 30, 300\}$, in the softmax function. In Fig. 13c, we compare the zero-shot performance of three variants of one-step FB on each domain. These results suggest that one-step FB is less sensitive to the choice of $\tau_{\text{reward}}$ on each domain. However, using larger values of $\tau_{\text{reward}}$ can slightly boost performance on OGBench domains. Therefore, we still tune the reward weighting temperature $\tau_{\text{reward}}$ for each domain separately and select the best candidates.

Finally, we study the effects of the representation dimension $d$. Both prior work [81, 104] and our Proposition 1 have suggested that the representation dimension plays an important role for one-step FB. We sweep over $d = \{25, 50, 100, 128, 256, 512\}$ and selectively show performance of several values. As shown in Fig. 13d, the choice of representation dimension $d$ can vary the performance of one-step FB significantly on both ExORL and OGBench domains. We find that using $d = 50$, which is the same value as in Park et al. [81] and Touati et al. [104], is sufficient for ExORL domains. However, when increasing the representation dimension, we do *not* observe a consistent improvement over zero-shot performance. We conjecture that a finite representation dimension $d < \infty$ always learns a low-rank approximation of the successor measure ratio as in Corollary 1. Thus, some choices of $d$ might result in a better low-rank approximation. On OGBench domains, we select $d = 512$ for consistency with prior work [5], although $d = 128$ gives better performance on some domains.

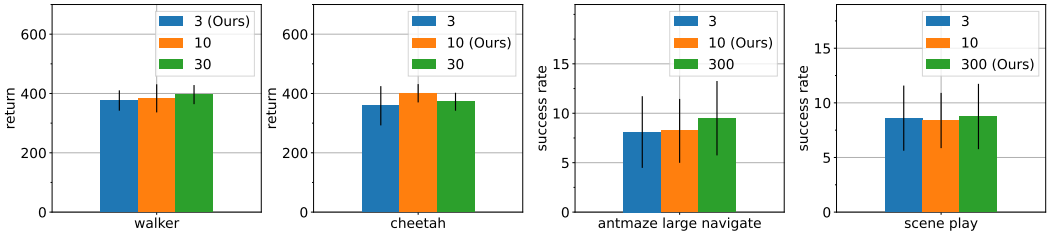
Taken together, we tune the behavioral-cloning regularization coefficient $\lambda_{\text{BC}}$, the orthonormalization coefficient $\lambda_{\text{ortho}}$, the reward weighting temperature $\tau_{\text{reward}}$, and the representation dimension $d$ on different domains. In general, our choices of hyperparameters are effective for one-step FB on both ExORL and OGBench domains.



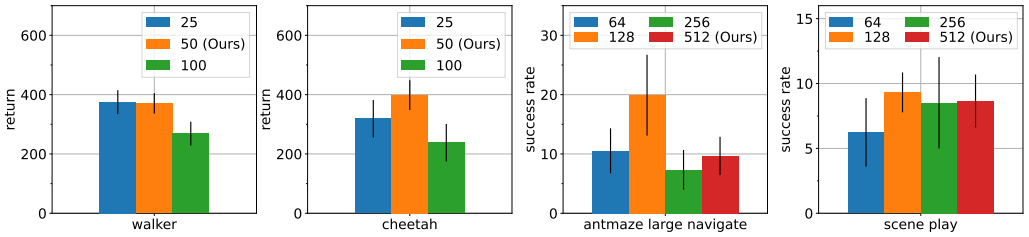

(a) Behavioral-cloning regularization coefficient $\lambda_{\text{BC}}$.




(b) Orthonormalization regularization coefficient $\lambda_{\text{ortho}}$.




(c) Reward weighting temperature $\tau_{\text{reward}}$.




(d) Representation dimension $d$.

Figure 13: **Hyperparameter ablations.** We conduct ablation experiments to study the effect of key components of one-step FB on `walker`, `cheetah`, `antmaze large navigate`, and `scene play`.