A SINGLE OCCUPANCY MEASURE IS OFTEN ENOUGH FOR ZERO-SHOT RL

Anonymous authors
Paper under double-blind review

ABSTRACT

As unsupervised pretraining becomes increasingly ubiquitous in reinforcement learning, developing a theoretical understanding of these methods is becoming as important as their empirical success. We focus on unsupervised learning via interaction, where forward-backward (FB) representation learning serves as a prototypical example. In standard FB, the policy defining the successor measure depends on the representation being learned, resulting in a coupled system. This raises the question: *Can the successor measure M^β of a single behavior policy suffice for zero-shot RL?* To understand the tradeoff this simplification introduces, we build on prior work that analyzes FB and the finite-dimensional factorizations of successor measures. We derive additional necessary conditions for exact FB representations, which we use to study learning behavior in didactic settings, and quantify the consequences of low-rank approximation used in practice. In didactic settings, we observed substantially improved convergence when learning M^β with $10^5\times$ smaller representation errors than FB. However, in a didactic counterexample, we observed that even perfect knowledge of M^β need not yield optimal downstream control. We then compare our simplified method, One-Step FB, to prior work across 10 state-based and image-based continuous control domains. One-Step FB yields representations achieving $1.4\times$ the zero-shot performance of FB on average. We further show that the zero-shot policies inferred by our algorithm provide an efficient initialization for further fine-tuning on downstream tasks.

1 INTRODUCTION

Large-scale pre-training has reshaped how we build learning systems in vision (Radford et al., 2021; Grill et al., 2020; Assran et al., 2023) and language (Team et al., 2023; Achiam et al., 2023): 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) (Tirinzoni et al., 2025; Sikchi et al., 2025; Li et al., 2025), 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.

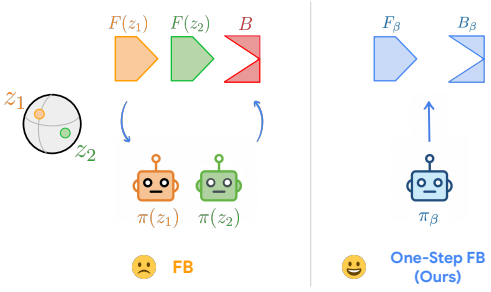


Figure 1: How can we learn representations that quickly adapt to new rewards? *(Left)* Prior work, FB (Touati & Ollivier, 2021), learns bilinear representations of successor measures for a family of latent-conditioned policies. *(Right)* We study whether representing only M^β of a single behavior policy suffices for zero-shot RL.

Forward-backward (FB) representation learning (Touati & Ollivier, 2021) 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 (Blier et al., 2021; Touati et al., 2022; Bagatella et al., 2025; Agarwal et al., 2025). In standard FB, however,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

the policy defining the successor measure depends on the representation being learned, resulting in a coupled learning system. In contrast, One-Step RL methods (Brandfonbrener et al., 2021; Gulcehre et al., 2020; Peters & Schaal, 2007; Wang et al., 2018; Peters et al., 2010; Eysenbach et al., 2022a) decouple Q-value estimation from policy extraction by first estimating the Q-values of the behavior policy and then performing a single policy-improvement step. Together, these suggest a simplification for zero-shot RL: learn the successor measure of a fixed behavior policy and defer policy improvement to downstream tasks. Broadly, giving rise to the central question of this paper:

Can the successor measure M^β of a single behavior policy suffice for zero-shot RL?

In this paper, we study the trade-off introduced by replacing standard FB with the successor measure M^β of a fixed behavior policy. First, we ask: *What additional necessary conditions must exact FB representations satisfy?* Answering this clarifies the requirements for exact FB, provides useful diagnostics in controlled experiments, and quantifies consequences of low-rank approximation. Second, we reinterpret the FB representation objective as a regression loss, revealing insights into *what the algorithm optimizes*. This connection to fitted Q-evaluation (FQE) (Ernst et al., 2005; Riedmiller, 2005; Fujimoto et al., 2022) motivates our study of its learning behavior. Third, we formalize the Bellman update implicit in the practical FB learning procedure. Unlike fixed-policy successor learning (Blier et al., 2021), the policy in this update is itself induced by the learned forward representation, creating a circular dependency.

Breaking the circular dependency (Fig. 1) connects FB to one-step RL, giving rise to a simplified method called **One-Step FB**. Our algorithm first learns representations for a *fixed* behavioral policy, thereby pre-training one step of policy improvement over all the behavioral value functions. In didactic experiments, One-Step FB exhibits improved convergence with $10^5\times$ smaller representation errors than FB. Across 8 state-based and 2 image-based benchmark domains, One-Step FB is a competitive method for unsupervised pre-training, achieving $1.4\times$ the zero-shot performance of FB on average. Our method also 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.

2 Preliminary

We first introduce the notation and background mathematics for our analysis, deferring a conceptual description of related work to Appendix A. We consider a controlled Markov process (CMP) (Bhatt & Borkar, 1996) 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) (Sutton et al., 1998). 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 (Dayan, 1993; Janner et al., 2020; Touati & Ollivier, 2021; Eysenbach et al., 2022c; Zheng et al., 2024a; Myers et al., 2024; Zheng et al., 2025) $M^\pi : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S} \times \mathcal{A})$ 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 (Dayan, 1993; Barreto et al., 2017; Blier et al., 2021) 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[1] 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., $\text{rank}(M^\pi) = |\mathcal{S} \times \mathcal{A}|$) (Agarwal et al., 2019, 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 (Brown et al., 2020; Ouyang et al., 2022). Next, we make this resemblance precise.

[1]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 (Lange et al., 2012) (Sec. 4), where learning happens on a dataset of transitions $\mathcal{D} = \{(s, a, s', a')\}$ collected by some behavioral policy $\pi_\beta : \mathcal{S} \to \Delta(\mathcal{A})$. Here, *zero-shot* RL denotes algorithms whose policy adaptation does not require updating the neural networks. See Appendix B.2 for additional details.

Forward-backward representation learning. FB is an instance of the zero-shot RL algorithms (Touati & Ollivier, 2021; Bagatella et al., 2025; Park et al., 2024b; Zheng et al., 2025; Agarwal et al., 2025). 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 $M^\pi(s_f, a_f \mid s, a, z)$. See Appendix B.3 for the formal definition of ground-truth representations $F^\star(s, a, z)$ and $B^\star(s_f, a_f)$.

Prior work (Touati & Ollivier, 2021; Touati et al., 2022) establishes the existence of exact FB representations in finite state-action spaces *when the representation dimension is sufficiently large*. We ask: *What additional necessary conditions must exact FB representations satisfy?* Sec. 3.1 answers this question using linear algebra and rank matching. Given exact representations, FB adapts to any reward function using $z_r = \mathbb{E}_{\rho(s_f, a_f)}[B^\star(s_f, a_f) r(s_f, a_f)]$, which indexes its optimal policy. We subsequently study practical FB learning (Sec. 3.3) and derive a simpler method (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 $\mathcal{X}$, when p is absolutely continuous with respect to q. The most widely adopted approach casts this problem as classification (Qin, 1998; Cheng & Chu, 2004; Gutmann & Hyvärinen, 2010). However, least-squares importance fitting (LSIF) (Kanamori et al., 2009; 2008) casts the measure ratio estimation as regression.[2] Specifically, LSIF fits a function $g : \mathcal{X} \to \mathbb{R}$ to the target measure ratio $p(x)/q(x)$ using samples.

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

where the constant is independent of the learned ratio $g(x)$. Unlike classification losses, 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 reinterpret the FB representation objective using the LSIF loss function.

3 Understanding Forward-Backward Representation Learning

In this section, we study FB through the lens of linear algebra, LSIF, and contraction mapping. The goal of our theoretical analysis is threefold.

- §3.1 We examine additional necessary constraints for exact FB representations arising from shared representations and positive affine transformations of reward. These conditions provide useful diagnostics and quantify the consequences of low-rank approximation.
- §3.2 We reinterpret the FB representation objective as a temporal-difference variant of a regression loss (Eq. 3), drawing a connection with FQE.
- §3.3 We formalize the Bellman update implicit in the learning procedure of practical FB and examine why the Banach fixed-point theorem (Banach, 1922) does not directly apply.

The main challenge lies in the circular dependency of FB (Fig. 1): both the latent-conditioned policy and its associated successor measure depend on the forward representation. We therefore introduce a variant of FB that breaks it (Sec. 4), albeit introducing a trade-off between a simpler representation learning objective and exact representations needed for optimal downstream control (Sec. 5).

3.1 What Necessary Constraints Must Exact FB Representations Satisfy?

Prior work has studied finite-rank factorizations of successor measures for fixed policies (Blier et al., 2021; Touati & Ollivier, 2021) and established the existence of exact FB representations in finite spaces (Proposition 6 of Touati & Ollivier (2021)). We ask a complementary question: *What additional necessary conditions must any exact FB representation satisfy?* We derive two sets of necessary conditions (Lemma 1): *(i)* exact FB requires shared factorization of the backward representation,

[2] The same loss recurs under different names in the literature (Nachum et al., 2019; Kato, 2025).

which imposes rank, dimensionality, and cross-policy consistency constraints (Proposition 1), and *(ii)* positive affine transformations of rewards impose invariance constraints on the forward representation and equivariance constraints on Q-functions (Proposition 2).

The first set of necessary conditions follows from the shared-factorization requirement of exact FB.

Proposition 1 (Informal)**.** *Given any discrete CMP, a finite latent space $\mathcal{Z}$, and a marginal measure ρ, 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 1 and Definition 2) must satisfy the following properties:*

1. ***Rank and dimensionality:*** *The representation dimension $d \geq |\mathcal{S}\times\mathcal{A}|$, $|\mathcal{S}\times\mathcal{A}| \leq rank(F^\star_Z) \leq d$ and $rank(B^\star) = |\mathcal{S}\times\mathcal{A}|$.*
2. ***Cross-policy consistency:*** *The matrix $B^\star$, components of matrix $F^\star_{\mathcal{Z}}$, the successor measures must satisfy: $B^\star = F^{\star+}_{z_i} M^{\pi(a|s,z_i)}/\rho + (I_d - F^{\star+}_{z_i}F^\star_{z_i})W_{z_i}\ \forall z_i \in \mathcal{Z}$, for arbitrary $W_{z_i} \in \mathbb{R}^{d\times|\mathcal{S}\times\mathcal{A}|}$. Here, X^+ denotes the pseudoinverse (Moore, 1920; Bjerhammar, 1951) of the matrix X. (See Appendix C.1 for the complete discussion.)*

Our rank analysis provides insight into the learnability of exact FB representations: the required representation dimension d is determined by the rank of the linear operator induced by the successor measure, independently of the expressive capacity of the neural networks.

Corollary 1 (Learnability; Informal)**.** *For a policy π_z, let its successor measure be M^{π_z} and the linear operator induced by M^{π_z} be defined as $(T_{M^{\pi_z}} r)(s,a) := \int_{\mathcal{S}\times\mathcal{A}} r(s_f,a_f) M^{\pi_z}(ds_f, da_f|s,a)$. Any exact d-dimensional FB representation requires $d \geq rank(T_{M^{\pi_z}})$. Hence, for finite CMPs, $d \geq |\mathcal{S}\times\mathcal{A}|$; for continuous CMPs, exact FB representations require $d = \infty$.*

See Appendix C.2 for the complete statement and the proof. The challenge not only stems from fitting the inner product (one can fit the inner product as a kernel function (Berlinet & Thomas-Agnan, 2011; Lu et al., 2019)), but also stems from the infinite-dimensional latent space when the successor operator has infinite-dimensional range. As mentioned in prior work (Touati & Ollivier, 2021; Touati et al., 2022), using a finite representation dimension results in a low-rank approximation of exact FB.

Additionally, 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 irreducible errors in the optimal Q-value predictions.

Corollary 2 (Irreducible Error; Informal)**.** *For a finite CMP with $d < |\mathcal{S}\times\mathcal{A}|$, any d-dimensional FB representation admits a normalized reward $\|r\|_\infty = 1$ for which $z_r = 0$. Consequently, its optimal Q-value prediction satisfies $\|\hat{Q}_r - Q^\star_r\|_\infty \geq \frac{1-\gamma}{1+\gamma}$. Thus, the low-dimensional FB representation incurs a non-zero irreducible error for some rewards.*

See Appendix C.3 for the complete statement and the proof. Unlike the function approximation errors in neural networks (Hornik et al., 1989), these errors in the optimal Q-value prediction are irreducible. Therefore, the corresponding zero-shot policy may not be optimal for maximizing the reward.

The second set of necessary conditions studies the exact forward representations under reward transformations. Recall that the Q-value is equivariant to an arbitrary positive affine transformation on the reward (Russell et al., 1995; Ng et al., 1999) (See Lemma 3). We translate this into an invariance property that the exact forward representations must satisfy:

Proposition 2 (Informal)**.** *Consider two rewards related by a positive affine transformation, $r' = \nu r + \xi$ where $\nu > 0$ and $\xi \in \mathbb{R}$. As r and r' induce the same optimal policy, their successor measures are the same. Hence, their forward representations are invariant modulo the kernel of the backward representation, i.e., $F^\star(s,a,z_{r'}) - F^\star(s,a,z_r) \in \ker(B^{\star T})$, and Q-values are equivariant, i.e., $Q^\star_{r'} = \nu Q^\star_r + \xi$. (See Appendix C.4 for further discussion and the proof.)*

This proposition underscores a *necessary* condition: any F that failed to satisfy Proposition 2 cannot equal $F^\star$ (contrapositive). Importantly, satisfying this condition alone is not sufficient to establish convergence. We use this proposition as a criterion in empirical evaluation of FB (Sec. 5.1).

3.2 What Does the FB Representation Objective Minimize?

We now reinterpret the representation objective used in FB. For the finite, discrete setting considered in our theoretical analysis, 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 & Ollivier (2021).

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:

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

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^π.

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 (Mnih et al., 2015). The resulting loss function minimizes a Bellman error [3]:

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

$$y = (1 - \gamma)\delta(s_f, a_f \mid s, a)/\rho(s_f, a_f) + \gamma \bar{F}(s', a', z)^\top \bar{B}(s_f, a_f).$$

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[4]:

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

$$- (1 - \gamma)\mathbb{E}_{p^{\pi_\beta}(s,a),\, p_{\mathcal{Z}}(z)} \left[F(s, a, z)^\top B(s, a) \right]. \tag{6}$$

See Appendix C.5 for the derivation. Importantly, we can now interpret the representation objective in FB as a fitted Bellman regression on the successor-measure ratio, which has a clear connection with the standard Bellman operator and the Banach fixed-point theorem (Banach, 1922). These theoretical connections motivate us to study whether FB admits a similar convergence guarantee in practice.

3.3 What Makes the Coupled Learning Dynamics of FB Challenging?

In this section, we identify a theoretical difficulty in FB arising from the dependence of the Bellman target in Eq. 5 on the greedy policy. Our analysis proceeds in two steps. *First*, the resemblance between Eq. 5 and FQE motivates us to define a new coupled FB Bellman target map. *Second*, we provide a concrete counterexample in a restricted setting where the Bellman update is a well-defined operator. We show that this operator can be discontinuous at greedy-policy switching boundaries and hence is not contractive in general. In contrast, fixing the behavior policy restores γ-contraction.

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, for each latent z, we can define a new *coupled FB Bellman target map* as

$$[\mathcal{H}(f, b)]_z(s, a, s_f, a_f) \triangleq (1 - \gamma)\frac{\delta(s_f, a_f \mid s, a)}{\rho(s_f, a_f)} + \gamma \mathbb{E}_{p(s'|s,a), \pi(a'|s',z)} \left[f(s', a', z)^\top b(s_f, a_f) \right]$$

Note that the map $\mathcal{H}$ takes as input two matrices and outputs another matrix $Y_z = \mathcal{H}_z(F_z, B)$. Eq. 5 requires one more step $\mathcal{R}_{\text{fit}}(Y_z) \in \arg\min_{F'_z, B'} \|{F'_z}^\top B' - Y_z\|^2$. Therefore, the complete update of F and B representations is given by $\mathcal{U} = \mathcal{R}_{\text{fit}} \circ \mathcal{H}$. Intuitively, $\mathcal{H}$ gives us the predicted target and $\mathcal{R}_{\text{fit}}$ updates the representations to approximate it.

The forward representation f plays two roles: *(i)* it determines the policy whose successor measure appears in the Bellman target $\mathcal{H}$, and *(ii)* it approximates Y_z in $\mathcal{R}_{\text{fit}}$ along with b. To isolate the policy dependence in $\mathcal{H}$ from $\mathcal{R}_{\text{fit}}$, we consider a restricted setting where $d = n, B = I_n, z = \text{diag}(\rho)r$ for a fixed reward $r \in \mathbb{R}^n$. In this setting, since $F_z^\top = Y_z$, the coupled target map reduces to a standard Bellman operator $\mathcal{T}_r(Y) = (1-\gamma)\text{diag}(\rho)^{-1} + \gamma P^{\pi_Y} Y$, where π_Y is the greedy policy with

[3] A similar formulation has been mentioned in prior work (See Appendix B of Touati et al. (2022)); here, we use the LSIF/FQE interpretation to motivate the fixed-point analysis in Sec. 3.3.

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

respect to $Q_Y = Y\text{diag}(\rho)r$ and P^{π_Y} is the policy-dependent transition matrix. Two arbitrarily close candidate matrices, Y^ϵ_+ and Y^ϵ_-, can rank actions in different orders and therefore induce different greedy policies. This results in their Bellman targets being separated by a non-zero distance even if Y^ϵ_+ and Y^ϵ_- approach a limiting value, creating a discontinuity at the switching boundary.

Proposition 3 (Informal)**.** *In this restricted setting, the Bellman update, $\mathcal{T}_r$, is not, in general, a γ-contraction under the entrywise supremum norm. Thus, the Banach fixed-point theorem cannot, in general, be used to establish the convergence of the coupled FB target update.*

See Appendix C.6 for the complete formal discussion. Our analysis does *not* suggest that the sequence of updates is non-convergent; it only invalidates a general convergence argument based on the Banach fixed-point theorem (Banach, 1922). On the other hand, if π_β is fixed, the corresponding update is given by $\mathcal{T}^\beta(Y) = (1-\gamma)\text{diag}(\rho)^{-1} + \gamma P^{\pi_\beta} Y$. As P^{π_β} is row-stochastic, $\|\mathcal{T}^\beta(Y) - \mathcal{T}^\beta(\hat{Y})\|_\infty \leq \gamma\|Y - \hat{Y}\|_\infty$. Thus, the fixed-policy update is a γ-contraction with a unique fixed point (See prior work (Dayan, 1993; Barreto et al., 2017; Blier et al., 2021) for the contraction proof).

This result motivates *One-Step FB*: fixing the behavior policy restores a standard contraction guarantee for the Bellman target in the restricted setting. The convergence of the complete update $\mathcal{U}$ in the general case remains an open question for future work.

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 (Park et al., 2024b; Zheng et al., 2025; Ghosh et al., 2023), 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_β and a backward representation function B_β to fit the fixed successor measure ratio of the behavioral policy $\pi_\beta(a \mid s)$. We use notations similar to FB, but F_β and B_β are semantically different from F and B, highlighting the dependency on the behavioral policy using the subscript β. We also introduce a new TD One-Step FB loss to learn the new FB representations:

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

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_β 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_β and B_β, enjoying stable convergence.

Similar to FB, we additionally regularize the backward representations to be orthonormal (Touati & Ollivier, 2021; Touati et al., 2022):

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

The complete new representation objective contains both the TD One-Step FB loss and the orthonormalization regularization, with λ_{ortho} controlling the regularization strength:

$$\mathcal{L}(F_\beta, B_\beta) = \mathcal{L}_{\text{TD One-Step FB}}(F_\beta, B_\beta) + \lambda_{\text{ortho}}\mathcal{L}_{\text{ortho}}(B_\beta). \tag{9}$$

In Appendix C.7, we discuss a connection between our method and the singular value decomposition (SVD) of the behavioral successor measure. We use the learned forward representations to derive a latent-conditioned policy in our algorithm.

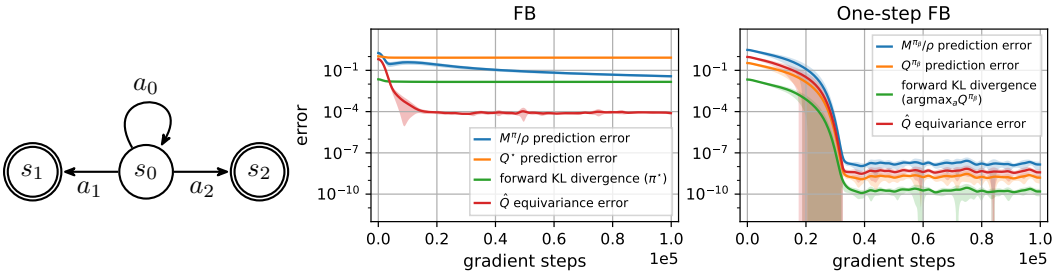


Figure 2: **Learning FB representations in a three-state CMP.** *(Left)* The agent starts at s_0 and action a_i deterministically transitions to s_i. States s_1 and s_2 are absorbing. Sec. 5.1 uses this CMP to study the convergence of FB and One-Step FB algorithms. *(Center)* After training for 10^5 gradient steps, FB fails to recover exact FB representations. *(Right)* Given a fixed policy, One-Step FB exactly fits the target representations within 4×10^4 gradient steps, illustrating simpler representation learning consistent with our theory (Sec. 3.3) and motivation for developing a new method (Sec. 4.1).

4.2 Learning a Policy Using Forward Representations

Our approach to learning a policy is similar to FB. Specifically, we select actions to maximize the inner products between the forward representation F_β and a latent variable z sampled from the latent prior $p_{\mathcal{Z}}$. For discrete CMPs, we use a softmax policy with temperature τ_{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 (Haarnoja et al., 2018) with an additional behavioral-cloning regularization (Fujimoto & Gu, 2021) for the offline setting, with λ_{BC} controlling the regularization strength:

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

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.8), similar to the generalized policy improvement in Barreto et al. (2017).

Algorithm summary. In Alg. 1, we summarize our new algorithm, **one-step forward-backward representation learning (One-Step FB)**. 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 use One-Step FB to study a trade-off: fixing the behavior policy simplifies representation learning but may limit downstream control. First, in a discrete CMP, we empirically examine the insights from Sec. 3 by comparing how well FB and One-Step FB learn their respective target representations. Second, another discrete-MDP counterexample shows a limitation of One-Step FB. Third, 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.

5.1 Can FB and One-Step FB Learn Their Target Representations?

We test whether FB and One-Step FB learn their respective target representations in a simple CMP (Fig. 2 *(Left)*) with three states and three actions. We chose this discrete CMP because we can compute the successor measure and the optimal Q-value analytically. Since FB and One-Step FB have different targets, we evaluate each method against its own target representations.

We train both methods for 10^5 gradient steps using their MC objectives and otherwise use the same optimization settings. We set $d = |\mathcal{S} \times \mathcal{A}|$ and use full-rank representations. We track several metrics (See Appendices D.1 and D.2 for implementation details and formal definitions) with the aim of answering the following questions:

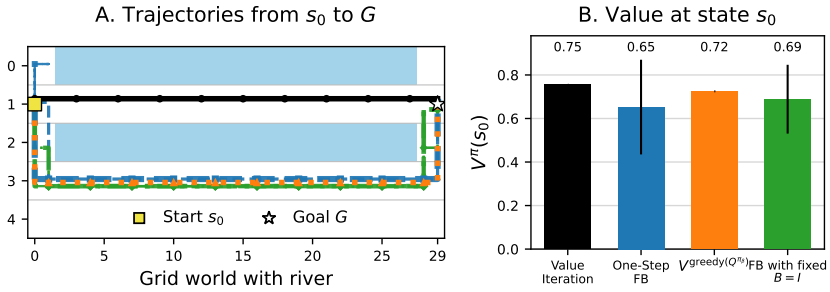


Figure 3: **One-Step FB need not yield optimal policies in the Drunk Spider Gridworld.** The environment has deterministic transitions with blue shaded cells indicating absorbing river states. *(Left)* Evaluation trajectories from s_0 to G for value iteration (black), analytical occupancy measure with one-step policy improvement (orange), FB with fixed $B = I$ (green), and One-Step FB (blue). *(Right)* Value at state s_0 with mean and standard deviation. One-Step FB chooses a suboptimal route with 33 steps, while value iteration chooses the shorter optimal route with 29 steps. Thus, One-Step FB and the analytical behavioral occupancy measure need not recover the optimal policy as obtained from value iteration.

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

FB does not recover its target representations. We train FB using the MC FB Loss (Eq. 4), which is the analytical analogue of the TD FB Loss (Eq. 6). As shown in Fig. 2*(Center)*, the errors plateau at high values, even asymptotically, suggesting that the FB algorithm did not converge. In particular, the successor measure ratio error approaches $\epsilon_{\text{SMR}} = 4 \times 10^{-2}$ (contrary to Definition 1). The policy KL divergence remains high at $\text{KL}_{\pi^\star} = 10^{-2}$, indicating that the FB algorithm failed to enable optimal policy adaptation (contrary to Definition 2). The predicted Q-values also violate affine equivariance with $\epsilon_{\text{equiv}} = 10^{-4}$. As every optimal Q-function satisfies this property, this violation provides evidence that FB has *not* recovered the target optimal Q-value. In Appendix E.3, we discuss potential confounding effects to clarify this observation.

One-Step FB recovers its fixed-policy target representations. 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. We train using a behavior policy π_β and the MC One-Step FB Loss (Eq. 27). Results in Fig. 2 *(Right)* suggest that all these metrics converge to small numbers ($\leq 10^{-7}$) within 4×10^4 gradient steps, helping to verify that One-Step FB recovers its target representations up to numerical precision. In particular, the learned One-Step FB representations are equivariant under an affine transformation in rewards ($\epsilon_{\text{equiv}} = 5 \times 10^{-9}$), which is consistent with the property of any Q-value (Lemma 3). This illustrates the learning benefit of removing the circular dependency in FB.

5.2 A Limitation of One-Step FB: Behavioral Successor Measures Need Not Yield Optimal Policies

One-Step FB learns the successor measure of a behavior policy π_β and performs one step of greedy policy improvement over $Q_r^{\pi_\beta}$. Although this guarantees policy improvement, it does not generally guarantee recovery of the optimal policy, which may require multiple policy improvement steps.

We illustrate this limitation using the deterministic gridworld in Fig. 3*(Left)* (See Appendix D.3 for implementation details). Value iteration follows the risky short direct route from s_0 to G and achieves the optimal value of 0.75 in 29 steps. In contrast, learned One-Step FB follows the longer suboptimal route, achieving a value of 0.65 in 33 steps on average. The greedy policy derived from analytically computed $Q_r^{\pi_\beta}$ also follows the longer route but achieves 0.72. FB with fixed $B = I$ also follows the longer suboptimal route with a predicted value of 0.69 on average. Thus, One-Step FB need not yield an optimal policy. Overall, these didactic experiments show the tradeoff introduced by One-Step FB.

Table 1: **Zero-shot evaluation on ExORL and OGBench benchmarks.** Reported scores are normalized by the maximum return (1000 for `walker`, `cheetah`, and `quadruped`; 250 for `jaco`) on ExORL and success rate (%) in OGBench. One-Step FB achieves the best performance on $5/10$ domains, outperforming FB by $1.4\times$ on average ($+-$). As in prior work (Park et al., 2025b), 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-γ	TD-JEPA	ICVF	HILP	FB	One-Step FB
`walker (4 tasks)`	22.8 ± 0.2	22.7 ± 0.2	31.6 ± 1.2	$\mathbf{61.9} \pm 2.3$	39.3 ± 10.8	40.0 ± 4.0	37.9 ± 2.6 (-2.1)
`cheetah (4 tasks)`	12.5 ± 4.1	12.7 ± 3.9	18.5 ± 0.7	18.7 ± 1.3	11.6 ± 7.8	27.1 ± 4.6	$\mathbf{37.8} \pm 5.6$ (+10.7)
`quadruped (4 tasks)`	46.2 ± 3.5	49.6 ± 3.5	46.1 ± 11.5	54.6 ± 3.7	35.2 ± 5.9	24.6 ± 3.1	$\mathbf{64.5} \pm 1.5$ (+39.9)
`jaco (4 tasks)`	1.2 ± 0.4	1.2 ± 0.0	5.2 ± 1.6	$\mathbf{9.2} \pm 1.2$	8.0 ± 2.0	4.0 ± 1.6	$\mathbf{8.8} \pm 1.6$ (+4.8)
`antmaze large navigate (5 tasks)`	9 ± 1	21 ± 2	14 ± 9	23 ± 3	$\mathbf{34} \pm 2$	25 ± 5	30 ± 9 (+5)
`antmaze teleport navigate (5 tasks)`	3 ± 1	16 ± 4	4 ± 1	$\mathbf{29} \pm 3$	19 ± 6	16 ± 8	11 ± 6 (-5)
`cube single play (5 tasks)`	6 ± 2	13 ± 2	8 ± 4	13 ± 2	$\mathbf{30} \pm 8$	2 ± 1	3 ± 2 (+1)
`scene play (5 tasks)`	4 ± 1	15 ± 8	6 ± 1	8 ± 6	$\mathbf{19} \pm 6$	6 ± 4	8 ± 2 (+2)
`visual cube single play (5 tasks)`	-	11 ± 4	13 ± 3	-	8 ± 1	12 ± 3	$\mathbf{14} \pm 3$ (+2)
`visual scene play (5 tasks)`	-	3 ± 1	11 ± 3	-	4 ± 1	13 ± 2	$\mathbf{16} \pm 4$ (+3)

5.3 How Does One-Step FB Compare to Prior Unsupervised RL Methods in Continuous-Control Domains?

We now compare One-Step FB to prior unsupervised RL algorithms, measuring the zero-shot adaptation performance on downstream tasks. While our didactic experiments show that One-Step FB enjoys simpler representation learning, one might wonder whether it suffices on standard benchmarks. Our experiments show that, empirically, it does. We defer the rationales for selecting prior methods to Appendix D.4. Appendix D.5 includes detailed discussions about environments, datasets, and evaluation protocols. Prior work (Park et al., 2024b; Bagatella et al., 2025; Agarwal et al., 2025) that evaluated on the same benchmarks reported inconsistent results. We therefore implement both our method and prior methods from scratch (See Appendix D.6 for additional details).

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 achieves the best or is within 95% of the best performance on 5 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 with 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 15% 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 explicitly learn 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 λ_{BC}, the orthonormalization regularization coefficient λ_{ortho}, the reward weighting temperature τ_{reward}, and the representation dimension d. As our main implementation intentionally uses a minimal training setup, we additionally implement One-Step FB in the released TD-JEPA (Bagatella et al., 2025) codebase for controlled comparison. In Appendix E.5, we report the originally published TD-JEPA results alongside our TD-JEPA rerun and One-Step FB results, together with the hyperparameters used.

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. Practically, the zero-shot policies inferred by One-Step FB can be suboptimal, similar to prior work (Park et al., 2024b; Zheng et al., 2025; Ghosh et al., 2023), if obtaining the optimal Q-value requires multiple steps of policy improvement.

AI USE STATEMENT

In this work, we used generative AI tools for grammar and spell checks, rephrasing, and assistance with editing code and launching experiments.

We did not use generative AI to develop theoretical results, write full sections, identify citations, or generate research ideas. Any AI-generated code was reviewed by atleast one author. The authors take full responsibility for the final content of this work, including text, mathematical statements and proofs, and experimental results.

ETHICS STATEMENT

Our work analyzes and studies a simplified prior zero-shot reinforcement learning algorithm in simulation using synthetic environments and public simulation benchmarks. It involves no human subjects, sensitive data, or real-world deployment. Therefore, it raises no immediate ethical concerns.

REPRODUCIBILITY STATEMENT

We will provide the source code for all our experiments upon publication. The experimental details are provided in Appendix D with hyperparameters for the main experiments given in Appendix D.6. The proofs for all theoretical results are provided in Appendix C.

REFERENCES

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.

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.

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.

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*, pp. 15619–15629, 2023.

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.

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

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.

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*, pp. 531–540. PMLR, 2018.

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

Abhay G Bhatt and Vivek S Borkar. Occupation measures for controlled Markov processes: Characterization and optimality. *The Annals of Probability*, pp. 1531–1562, 1996.

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.

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

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.

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.

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.

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.

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.

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 (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=a7APmM4B9d.

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

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*, pp. 1953–1963. PMLR, 2021.

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

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 (eds.), *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.

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

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*, pp. 1407–1416. PMLR, 2018.

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.

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.

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*, 2022a. URL https://openreview.net/forum?id=GXiWE8kDTcn.

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 (eds.), *Advances in Neural Information Processing Systems*, 2022b. URL https://openreview.net/forum?id=iqCO3jbPjYF.

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, 2022c.

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

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

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

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.

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

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.

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.

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.

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*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

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*, pp. 1861–1870. PMLR, 2018.

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

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

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.

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.

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.

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.

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

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

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

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, pp. 128–135. IEEE, 2005.

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

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

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

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

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.

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

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.

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

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

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.

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.

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

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.

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.

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

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.

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

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.

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.

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

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.

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

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.

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.

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

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.

Seohong Park, Oleh Rybkin, and Sergey Levine. METRA: Scalable unsupervised RL with metric-aware abstraction. *arXiv preprint arXiv:2310.08887*, 2023.

Seohong Park, Kevin Frans, Benjamin Eysenbach, and Sergey Levine. OGBench: Benchmarking offline goal-conditioned RL. *arXiv preprint arXiv:2410.20092*, 2024a.

Seohong Park, Tobias Kreiman, and Sergey Levine. Foundation policies with Hilbert representations. In *International Conference on Machine Learning*, pp. 39737–39761. PMLR, 2024b.

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

Seohong Park, Qiyang Li, and Sergey Levine. Flow Q-learning. *arXiv preprint arXiv:2502.02538*, 2025b.

Seohong Park, Aditya Oberai, Pranav Atreya, and Sergey Levine. Transitive RL: Value learning via divide and conquer. *arXiv preprint arXiv:2510.22512*, 2025c.

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*, pp. 745–750, 2007.

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

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*, pp. 5171–5180. PMLR, 2019.

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

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*, pp. 8748–8763. PmLR, 2021.

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

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

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

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

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.

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.

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.

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

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

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.

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.

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.

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*, pp. 5026–5033. IEEE, 2012.

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

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

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.

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

Yifan Wu, George Tucker, and Ofir Nachum. The Laplacian in RL: Learning representations with efficient approximations. *arXiv preprint arXiv:1810.04586*, 2018.

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.

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.

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.

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)*, pp. 2371–2378. IEEE, 2017.

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.

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

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*, 2024b.

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

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=HN0CYZbAPw.

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-γ	TD-JEPA	ICVF	HILP	FB	One-Step FB
walker	**overall**	22.8 ± 0.2	22.7 ± 0.2	31.6 ± 1.2	**61.9** ± 2.3	39.3 ± 10.8	40.0 ± 4.0	37.9 ± 2.6 (-2.1)
	flip	24.3 ± 0.5	24.2 ± 0.4	38.1 ± 3.4	**53.8** ± 1.9	33.2 ± 13.5	27.7 ± 10.0	38.8 ± 3.0
	run	8.9 ± 0.1	8.9 ± 0.2	11.2 ± 2.2	**25.8** ± 2.0	13.6 ± 3.6	19.4 ± 2.0	19.8 ± 3.0
	stand	38.9 ± 0.3	38.7 ± 0.5	38.7 ± 1.1	**85.8** ± 3.1	69.1 ± 12.6	62.1 ± 9.3	52.4 ± 6.0
	walk	19.2 ± 0.4	19.2 ± 0.4	38.4 ± 4.7	**82.1** ± 4.0	41.3 ± 19.5	50.6 ± 3.4	40.7 ± 6.3
cheetah	**overall**	12.5 ± 4.1	12.7 ± 3.9	18.5 ± 0.7	18.7 ± 1.3	11.6 ± 7.8	27.1 ± 4.6	**37.8** ± 5.6 (+10.7)
	run	4.0 ± 1.3	4.2 ± 1.3	6.6 ± 0.3	8.9 ± 0.7	3.8 ± 3.2	4.3 ± 3.6	**9.7** ± 1.9
	run backward	5.0 ± 1.7	4.8 ± 1.5	6.9 ± 0.6	4.8 ± 0.9	3.6 ± 4.0	12.5 ± 3.0	**18.1** ± 6.0
	walk	18.5 ± 6.1	19.9 ± 6.1	29.3 ± 1.5	38.5 ± 2.1	19.5 ± 12.1	25.1 ± 16.6	**42.4** ± 4.4
	walk backward	22.6 ± 7.7	22.0 ± 7.0	31.2 ± 2.6	22.6 ± 3.8	19.4 ± 20.2	66.3 ± 14.3	**81.1** ± 15.6
quadruped	**overall**	46.2 ± 3.5	49.6 ± 3.5	46.1 ± 11.5	54.6 ± 3.7	35.2 ± 5.9	24.6 ± 3.1	**64.5** ± 1.5 (+39.9)
	jump	55.4 ± 5.4	60.3 ± 6.7	49.4 ± 17.5	61.7 ± 5.9	32.1 ± 6.3	24.7 ± 8.4	**70.7** ± 4.8
	run	32.4 ± 2.2	34.5 ± 1.9	36.3 ± 11.2	39.5 ± 3.3	27.7 ± 5.6	16.5 ± 5.1	**46.1** ± 1.1
	stand	65.1 ± 4.7	70.0 ± 4.1	78.5 ± 17.1	79.6 ± 5.7	47.3 ± 10.3	38.8 ± 8.6	**91.6** ± 3.0
	walk	31.8 ± 2.1	33.7 ± 1.3	20.1 ± 10.0	37.5 ± 5.7	33.9 ± 11.1	18.3 ± 5.9	**49.6** ± 2.7
jaco	**overall**	1.2 ± 0.4	1.2 ± 0.0	5.2 ± 1.6	**9.2** ± 1.2	8.0 ± 2.0	4.0 ± 1.6	8.8 ± 1.6 (+4.8)
	reach bottom left	1.2 ± 0.4	1.2 ± 0.4	4.0 ± 1.2	10.0 ± 3.6	**11.6** ± 3.6	3.2 ± 2.0	2.4 ± 0.8
	reach bottom right	1.2 ± 0.0	1.2 ± 0.4	4.0 ± 2.0	8.4 ± 3.2	**9.6** ± 3.2	2.8 ± 3.6	2.0 ± 0.8
	reach top left	0.8 ± 0.4	1.2 ± 0.0	6.8 ± 2.4	10.4 ± 4.0	2.4 ± 3.2	3.6 ± 3.6	**21.2** ± 4.4
	reach top right	1.6 ± 0.4	1.2 ± 0.4	6.4 ± 2.8	8.0 ± 2.8	**8.8** ± 4.4	6.8 ± 3.6	**8.8** ± 2.4
antmaze large navigate	**overall**	9 ± 1	21 ± 2	14 ± 9	23 ± 3	**34** ± 2	25 ± 5	30 ± 9 (+5)
	task 1	2 ± 1	6 ± 3	15 ± 11	4 ± 2	13 ± 8	**46** ± 9	21 ± 9
	task 2	2 ± 1	11 ± 4	1 ± 2	9 ± 3	16 ± 6	2 ± 3	**41** ± 12
	task 3	29 ± 3	57 ± 8	35 ± 23	67 ± 8	**75** ± 6	31 ± 10	15 ± 4
	task 4	6 ± 2	14 ± 5	7 ± 7	18 ± 6	27 ± 10	3 ± 2	**33** ± 15
	task 5	6 ± 3	16 ± 4	11 ± 8	18 ± 4	40 ± 8	**44** ± 19	37 ± 20
antmaze teleport navigate	**overall**	3 ± 1	16 ± 4	4 ± 1	**29** ± 3	19 ± 6	16 ± 8	11 ± 6 (-5)
	task 1	1 ± 1	5 ± 3	0 ± 0	**17** ± 9	10 ± 5	8 ± 9	2 ± 1
	task 2	6 ± 2	15 ± 7	2 ± 1	**38** ± 17	27 ± 7	11 ± 10	15 ± 10
	task 3	4 ± 2	15 ± 3	7 ± 3	**40** ± 4	24 ± 8	23 ± 10	17 ± 10
	task 4	2 ± 1	24 ± 12	7 ± 2	**40** ± 6	21 ± 11	25 ± 10	19 ± 11
	task 5	2 ± 1	**23** ± 5	2 ± 2	12 ± 11	14 ± 7	13 ± 12	2 ± 1
cube single play	**overall**	6 ± 2	13 ± 2	8 ± 4	13 ± 2	**30** ± 8	2 ± 1	3 ± 2 (+1)
	task 1	5 ± 2	12 ± 4	6 ± 4	13 ± 2	**27** ± 7	1 ± 1	3 ± 4
	task 2	5 ± 2	13 ± 3	8 ± 6	13 ± 3	**30** ± 10	3 ± 2	4 ± 4
	task 3	7 ± 3	13 ± 4	14 ± 9	14 ± 4	**23** ± 13	3 ± 3	4 ± 4
	task 4	6 ± 2	15 ± 6	6 ± 4	11 ± 3	**37** ± 19	2 ± 2	2 ± 2
	task 5	5 ± 3	11 ± 3	5 ± 4	13 ± 4	**30** ± 22	1 ± 2	1 ± 1
scene play	**overall**	4 ± 1	15 ± 8	6 ± 1	8 ± 6	**19** ± 6	6 ± 4	8 ± 2 (+2)
	task 1	17 ± 6	49 ± 32	30 ± 6	34 ± 23	**66** ± 16	25 ± 16	21 ± 8
	task 2	1 ± 1	11 ± 8	0 ± 0	5 ± 7	**14** ± 11	5 ± 5	12 ± 4
	task 3	1 ± 1	9 ± 6	0 ± 0	3 ± 3	**12** ± 14	0 ± 1	0 ± 0
	task 4	2 ± 1	4 ± 7	0 ± 0	0 ± 0	1 ± 1	0 ± 0	**7** ± 4
	task 5	**0** ± **0**	**0** ± **0**	**0** ± **0**	**0** ± **0**	**0** ± **0**	**0** ± **0**	**0** ± **0**
visual cube single play	**overall**	-	11 ± 4	13 ± 3	-	8 ± 1	12 ± 3	**14** ± 3 (+2)
	task 1	-	24 ± 16	19 ± 7	-	10 ± 6	14 ± 7	**47** ± 12
	task 2	-	10 ± 2	13 ± 8	-	**19** ± 5	10 ± 5	11 ± 3
	task 3	-	**16** ± 4	**16** ± 7	-	10 ± 7	15 ± 6	3 ± 3
	task 4	-	3 ± 3	6 ± 5	-	**13** ± 7	8 ± 5	9 ± 8
	task 5	-	1 ± 1	10 ± 5	-	**15** ± 3	11 ± 7	2 ± 2
visual scene play	**overall**	-	3 ± 1	11 ± 3	-	4 ± 1	13 ± 2	**16** ± 4 (+3)
	task 1	-	7 ± 2	53 ± 16	-	14 ± 4	66 ± 12	**78** ± 16
	task 2	-	1 ± 1	0 ± 1	-	2 ± 1	0 ± 0	**3** ± 4
	task 3	-	0 ± 1	0 ± 0	-	**2** ± 1	0 ± 0	0 ± 0
	task 4	-	**5** ± 5	0 ± 0	-	0 ± 0	0 ± 0	1 ± 1
	task 5	-	**0** ± 1	**0** ± **0**	-	**0** ± 1	**0** ± **0**	**0** ± **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 (Touati & Ollivier, 2021).

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 (Frans et al., 2024; Park et al., 2024b; Kim et al., 2024; Hu et al., 2023; Park et al., 2023; Eysenbach et al., 2018; Zheng et al., 2024b), intent predictions (Frans et al., 2024; Zheng et al., 2025), empowerment maximization (Klyubin et al., 2005; Choi et al., 2021; Mohamed & Jimenez Rezende, 2015; Shah et al., 2025), or self-supervised representation learning (Ni et al., 2024; Ma et al., 2023; Mazoure et al., 2023; Zhang et al., 2021; Zheng et al., 2023). 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 (Park et al., 2023; Frans et al.,

Algorithm 1 One-step forward-backward representation learning

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

2024; Ma et al., 2023; Zheng et al., 2025). One appealing family of methods that does not require gradient-based fine-tuning of the pre-trained policy is called zero-shot RL methods (Touati & Ollivier, 2021; Park et al., 2024b; Bagatella et al., 2025; Tirinzoni et al., 2025; Li et al., 2025). Similar to the in-context learning in LLMs (Brown et al., 2020), 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 (Touati & Ollivier, 2021)) and study its convergence in practice.

Successor measures and successor features. Our work builds on successor measures (Dayan, 1993), which were originally proposed to improve generalization in RL and have since been widely adopted in neuroscience to model predictive maps in the brain (Momennejad et al., 2017; Xie et al., 2025). In the domain of Deep RL, prior work has shown that successor measures can be learned in high-dimensional environments (Kulkarni et al., 2016; Zhang et al., 2017) and facilitate transfer learning across tasks (Barreto et al., 2017). By combining these ideas with universal value function approximators (Schaul et al., 2015), Universal Successor Features (USFs) generalize successor features to estimate values for any reward under any policy (Borsa et al., 2019). More recently, forward-backward (FB) representation learning (Touati & Ollivier, 2021) extended this to enable zero-shot evaluation for *any* reward function, forming the basis for building behavioral foundation models (Tirinzoni et al., 2025; Bagatella et al., 2025). 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 (Lopez-Paz & Oquab, 2016), covariate shift adaptation (Sugiyama et al., 2008), outlier detection (Sugiyama et al., 2008), mutual information estimation (Belghazi et al., 2018; Poole et al., 2019), and policy evaluation (Nachum et al., 2019). Prior work has tackled the density ratio estimation problem by minimizing a KL divergence (Sugiyama et al., 2008), moment matching (Gretton et al., 2009), penalized convex risk minimization (Nguyen et al., 2010), contrastive learning (Ma & Collins, 2018; Oord et al., 2018; Poole et al., 2019). Our analysis of the FB algorithm is closely related to the least-squares importance fitting approach (Kanamori et al., 2008; 2009) 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) Riedmiller (2005).

One-step RL. One-step RL methods Brandfonbrener et al. (2021); Gulcehre et al. (2020); Peters & Schaal (2007); Wang et al. (2018); Peters et al. (2010) 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 Peters et al. (2010)

to goal-conditioned imitation learning Savinov et al. (2018); Ding et al. (2019); Li et al. (2020); Chen et al. (2021); Kumar et al. (2019); Eysenbach et al. (2020; 2022b); Wantlin et al. (2025). Theoretical and empirical analysis presented in Eysenbach et al. (2022a) 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 Eysenbach et al. (2022c); Park et al. (2025c); Brandfonbrener et al. (2021) 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 π, 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. (2019))**.** *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^π 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^∞-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^∞-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 (Frans et al., 2024) or a scaled von Mises-Fisher distribution (Park et al., 2024b; Touati & Ollivier, 2021), 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 (Park et al., 2024b; Touati & Ollivier, 2021; Agarwal et al., 2025; Bagatella et al., 2025). 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. We refer to these ground-truth representations as exact FB representations.

Definition 1 (Definition 1 of Touati & Ollivier (2021))**.** *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 2 (heorem 2 of Touati & Ollivier (2021))**.** *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 (Touati & Ollivier, 2021; Touati et al., 2022; Bagatella et al., 2025) 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 Rank and Dimensionality Constraints 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 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 consider $|\mathcal{Z}|$ as a finite number and take the limit to infinity ($|\mathcal{Z}| \to \infty$). We 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^π'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:

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

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 1 and the optimal latent adaptation in Definition 2 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 $\text{diag}(\rho) \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|\times|\mathcal{S}\times\mathcal{A}|}$ is the diagonal matrix of the marginal measure vector ρ 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 $\text{rank}\,(F_\mathcal{Z})$, the rank of the backward representation matrix $\text{rank}(B)$, and the rank of the product of the forward-backward representation matrices $\text{rank}\,(F_\mathcal{Z}B)$ by matrix dimensions.

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

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* $F_\mathcal{Z}^\star \in \mathbb{R}^{|\mathcal{Z}\times\mathcal{S}\times\mathcal{A}|\times d}$ *(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* $M_\mathcal{Z}^\pi$ *(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. *$F_\mathcal{Z}^\star$ and $B^\star$ fit the successor measure ratio (Definition 1):*

$$M_\mathcal{Z}^\pi \text{diag}(\rho)^{-1} = F_\mathcal{Z}^\star B^\star, \tag{21}$$

2. *$F_\mathcal{Z}^\star$ and $B^\star$ enable optimal policy adaptation (Definition 2): 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 $\pi_r^\star(a \mid s) = \pi(a \mid s, z_r) = \arg\max_a F^\star(s, a, z_r)^\top z_r$ and the optimal Q-value $Q_r^\star = Q_{z_r} = F_{z_r}^\star z_r$,

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

$$B^\star = F_i^{\star+} M_i^\pi \text{diag}(\rho)^{-1} + (I_d - F_i^{\star+} F_i^\star) W_i$$

for arbitrary $W_i \in \mathbb{R}^{d\times|\mathcal{S}\times\mathcal{A}|}$. Here, X^+ denotes the pseudoinverse (Moore–Penrose inverse) (Moore, 1920; Bjerhammar, 1951) of the matrix X and $\text{diag}(x)$ is the diagonal matrix of the vector x.

In the special case of $d = |\mathcal{S}\times\mathcal{A}|$, each $F_{z_i}^{\star+}$ is invertible and $F_{z_i}^{\star+} F_{z_i}^\star = I_d$, giving us

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

Proof. The main idea of our proof is to match the rank of the FB representation matrices $F_\mathcal{Z}^\star, B^\star$ to the rank of the successor measure matrix $M_\mathcal{Z}^\pi$.

Rank matching. Since we aim to find a forward representation matrix $F_{\mathcal{Z}}^{\star}$ 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:

$$\text{rank}\left(M_{\mathcal{Z}}^{\pi}\right) = \text{rank}\left(M_{\mathcal{Z}}^{\pi}\text{diag}(\rho)^{-1}\right) = \text{rank}(F_{\mathcal{Z}}^{\star}B^{\star}).$$

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

$$\text{rank}(M_{\mathcal{Z}}^{\pi}) = \text{rank}\left(F_{\mathcal{Z}}^{\star}B^{\star}\right) = |\mathcal{S} \times \mathcal{A}|.$$

We next constrain the rank of the forward representation matrix $F_{\mathcal{Z}}^{\star}$ 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

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

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 $F_{\mathcal{Z}}^{\star}B$ 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, $F_i^{\star}$ (Eq. 16), and successor measure, M_i^{π} (Eq. 18), associated with each latent. 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 ρ does not change the column space of M_i^π because ρ has the full support over $\mathcal{S} \times \mathcal{A}$:

$$\text{col}\left(M_i^\pi \text{diag}(\rho)^{-1}\right) = \text{col}\left(M_i^\pi\right) \;;\; \text{rank}(M_i^\pi \text{diag}(\rho)^{-1}) = |\mathcal{S} \times \mathcal{A}|$$

Therefore,

$$|\mathcal{S} \times \mathcal{A}| = \text{rank}(F_i^\star B^\star) \leq \min(\text{rank}(F_i^\star), \text{rank}(B^\star))$$

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 $i = 1, \cdots, |\mathcal{Z}|$,

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

Similarly, together with dimension of $B^\star$, we get, $\text{rank}(B^\star) = |\mathcal{S} \times \mathcal{A}|$. 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:

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

This result shows that each $F_i^\star$ has full row rank and B^* has full column rank. Therefore, by the definition of the pseudoinverse of a matrix, all solutions to $F_i^\star B = M_i^\pi \text{diag}(\rho)^{-1}$ take the form:

$$B = F_i^{\star+} M_i^\pi \text{diag}(\rho)^{-1} + (I_d - F_i^{\star+} F_i^\star) W_i$$

for arbitrary $W_i \in \mathbb{R}^{d\times|\mathcal{S}\times\mathcal{A}|}$. Here, $F_i^{\star+}$ is the pseudoinverse of matrix $F_i^\star$. Also recall from linear algebra, that if solutions exist but $F_i^\star$ does not have full column rank, then we have indeterminate system, all of whose infinitude of solutions are given by this equation.

These intriguing observations help us find the additional conditions necessary for exact FB representation matrices: the exact backward representation matrix must be a *common* $B^\star$ that belongs to these solution sets for every latent variable z_i.

When $d = |\mathcal{S} \times \mathcal{A}|$, each $F_i^\star$ is square and invertible, resulting in $F_i^{\star+} F_i^\star = I_d$, and therefore,

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

When $d > |\mathcal{S} \times \mathcal{A}|$, $F_i^{\star+} F_i^\star \neq I_d$, so the backward representation is not uniquely determined by $F_i^\star$ and $M_i^\pi \text{diag}(\rho)^{-1}$.

Importantly, the exact FB representation matrices are not unique even for $d = |\mathcal{S} \times \mathcal{A}|$. For example, for any orthogonal matrix $Q \in \mathbb{R}^{d\times d}$, replacing $F_i^\star \mapsto F_i^\star Q^T$ and $B^\star \mapsto QB^\star$, leaves the inner product $F_i^\star B^\star$ unchanged.

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 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 Learnability of Exact FB Representations

Corollary 1. *For a latent variable z, let π_z denote the corresponding latent-conditioned policy and let M^{π_z} denote its successor measure. The linear operator induced by its successor measure is defined as,*

$$(T_{M^{\pi_z}} r)(s, a) := \int_{\mathcal{S}\times\mathcal{A}} r(s_f, a_f) M^{\pi_z}(ds_f, da_f | s, a).$$

which maps a reward function r to its action-value function $Q_r^{\pi_z}$. We define the rank of this operator as

$$\textit{rank}(T_{M^{\pi_z}}) := \dim(\textit{range}(T_{M^{\pi_z}}))$$

Any exact d-dimensional FB representation must satisfy $d \geq \textit{rank}(T_{M^{\pi_z}})$ for every latent-conditioned policy π_z. Consequently, for finite CMPs, $d \geq |\mathcal{S} \times \mathcal{A}|$; for continuous CMPs where $T_{M^{\pi_z}}$ has infinite-dimensional range, exact FB representations require an infinite-dimensional representation space, $d = \infty$.

Proof. For a fixed latent-conditioned policy π_z, exact FB implies that for every reward function r,

$$(T_{M^{\pi_z}} r)(s, a) := F^\star(s, a, z)^T \int_{\mathcal{S}\times\mathcal{A}} B^\star(s_f, a_f) r(s_f, a_f) \rho(ds_f, da_f)$$

The integral maps to $z_r \in \mathbb{R}^d$. Hence as r varies, every function in range$(T_{M^{\pi_z}})$ lies in the span of the d coordinate functions of $F^\star(\cdot, \cdot, z)$. Therefore,

$$\text{rank}(T_{M^{\pi_z}}) \leq d$$

For finite CMPs, Proposition 1 shows that

$$\text{rank}(T_{M^{\pi_z}}) = \text{rank}(M^{\pi_z}) = |\mathcal{S} \times \mathcal{A}|$$

and the exact factorization satisfies $d \geq |\mathcal{S} \times \mathcal{A}|$.

For continuous CMPs, FB aims support arbitrary reward functions $r : \mathcal{S}\times\mathcal{A} \to \mathbb{R}$. Since a continuous state-action space, $\mathcal{S} \times \mathcal{A}$, contains infinitely many distinct state-action pairs, the space of reward functions is infinite-dimensional. We denote this infinite-dimensional reward function space by $\mathcal{R}$.

For a fixed latent-conditioned policy π_z, $T_{M^{\pi_z}}$ maps each reward function $r \in \mathcal{R}$ to its corresponding action-value function. Additionally, this linear operator satisfies the Bellman equation

$$T_{M^{\pi_z}} r = (1 - \gamma) r + \gamma P^{\pi_z} T_{M^{\pi_z}} r$$

Suppose $T_{M^{\pi_z}} r = 0$. Substituting into the above equation gives us $0 = (1 - \gamma) r$. Since $\gamma < 1$, $r = 0$. Thus, $T_{M^{\pi_z}}$ is linear and injective, it preserves linear independence: every collection of linearly independent rewards is mapped to a collection of linearly independent action-value functions. Since, $\mathcal{R}$ is infinite-dimensional,

$$\begin{aligned} &\text{range}(T_{M^{\pi_z}}) \simeq \mathcal{R} \\ \implies &\text{rank}(T_{M^{\pi_z}}) = \infty \end{aligned}$$

This observation together with rank$(T_{M^{\pi_z}}) \leq d$, implies that no finite d is sufficient. Thus, exact FB representations require an infinite-dimensional representation space, $d = \infty$. □

C.3 Low-Rank Approximation Incurs Irreducible Errors

Corollary 2. *Given any finite CMP, a 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 $M_{\mathcal{Z}}^{\pi} \textit{diag}(\rho)^{-1} \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$.*

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, there exists a normalized reward vector $\|r_{null}\|_\infty = 1$, $r_{null} \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$ such that $z_{null} = 0$ and

$$\epsilon(r_{null}) \geq \frac{1-\gamma}{1+\gamma}$$

Thus, when $d < |\mathcal{S} \times \mathcal{A}|$, a d-dimensional FB representation incurs a nonzero irreducible optimal Q-value prediction error for some normalized reward.

Proof. Let $n = |\mathcal{S} \times \mathcal{A}|$. Proposition 1 shows that an exact FB representation requires the backward representation matrix to have rank $\text{rank}(B) = n$, and consequently require $d \geq n$. Therefore, when $d < n$, the d- dimensional FB matrices produce a low-rank approximation on the successor measure ratio:

$$\text{rank}(F_{\mathcal{Z}} B) \leq d < n = \text{rank}(M^\pi_{\mathcal{Z}} \text{diag}(\rho)^{-1}).$$

since ρ has full support and $\text{diag}(\rho)$ is invertible, and $\text{rank}(M^\pi_{\mathcal{Z}}) = n$ In this case, the rank of the backward representation matrix satisfies

$$\begin{aligned} &\text{rank}(B) \leq \min(d, n) = d < n \\ \implies &\text{rank}(B\text{diag}(\rho)) = \text{rank}(B) = d < n \end{aligned}$$

By the rank-nullity theorem, there exists a non-zero reward vector $r_{\text{null}} \in \mathbb{R}^n$ such that

$$B\text{diag}(\rho) r_{\text{null}} = 0$$

Because the null space is closed under scalar multiplication with $c > 0$, we can normalize r_{null} such that $\|r_{\text{null}}\|_\infty = 1$. We consider only normalized rewards so that the error bound is non-trivial. The latent variable induced by this reward is

$$z_{r_{\text{null}}} = B(r_{\text{null}} \odot \rho) = 0$$

Consequently, the optimal Q-value prediction is

$$\hat{Q}_{r_{\text{null}}} = F_{z_{r_{\text{null}}}} z_{r_{\text{null}}} = 0$$

We next lower bound the norm of the true optimal Q-value inorder to derive the irreducible error. Let $\pi^\star_{r_{\text{null}}}$ be the optimal policy for r_{null}. The optimal Q-value satisfies Bellman optimality

$$Q^\star_{r_{\text{null}}} = (1-\gamma) r_{\text{null}} + \gamma P^{\pi^\star_{r_{\text{null}}}} Q^\star_{r_{\text{null}}}$$

where $P^{\pi^\star_{r_{\text{null}}}}$ is the row stochastic transition matrix induced by the policy. Rearranging terms, taking the infinity norm on both sides and applying the triangle inequality gives us

$$(1-\gamma)\|r_{\text{null}}\|_\infty \leq \|Q^\star_{r_{\text{null}}}\|_\infty + \gamma \|P^{\pi^\star_{r_{\text{null}}}} Q^\star_{r_{\text{null}}}\|_\infty$$

Since $\|P^{\pi^\star_{r_{\text{null}}}}\| = 1$, $\|r_{\text{null}}\|_\infty = 1$ and rearranging terms,

$$\|Q^\star_{r_{\text{null}}}\|_\infty \geq \frac{1-\gamma}{1+\gamma}$$

As $\hat{Q}_{r_{\text{null}}} = 0$,

$$\epsilon(r_{\text{null}}) = \|Q^\star_{r_{\text{null}}} - \hat{Q}_{r_{\text{null}}}\|_\infty = \|Q^\star_{r_{\text{null}}}\|_\infty \geq \frac{1-\gamma}{1+\gamma}$$

Lastly, note that the general result for $\|r_{\text{null}}\|_\infty = c, c > 0$, is

$$\epsilon(r_{\text{null}}) \geq c \left(\frac{1-\gamma}{1+\gamma}\right)$$

Thus, when $d < |\mathcal{S} \times \mathcal{A}|$, the low-dimensional FB representation necessarily incurs a non-zero optimal Q-value prediction error for at least one normalized reward. □

C.4 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 then use this property to derive the invariance property of the ground-truth forward representations in FB.

Lemma 3 (Equivariance of Q-value under affine transformations)**.** *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 $r' = \nu r(s, a) + \xi$. Then, the Q-values of the reward functions r and r' satisfies $Q^\pi_{r'}(s, a) = \nu Q^\pi_r(s, a) + \xi$.*

Proof. Using equation 2,

$$Q^\pi_{r'}(s, a) = \mathbb{E}_{(s_f, a_f) \sim M^\pi(s_f, a_f | s, a)}[\nu r(s, a) + \xi]$$

Noting linearity of conditional expectation and because $M^\pi(\cdot \mid s, a)$ is a probability measure,

$$Q^\pi_{r'}(s, a) = \nu \mathbb{E}_{(s_f, a_f) \sim M^\pi(s_f, a_f | s, a)}[r(s, a)] + \xi = \nu Q^\pi_r(s, a) + \xi$$

□

Now, for the ground-truth forward representations, we also have the invariance 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 $r' = \nu r(s, a) + \xi$. Then, z_r and $z_{r'}$ are the latent variables indexing the optimal Q-value for the rewards r and r', respectively. Furthermore, r and r' induce the same optimal policy and hence the same successor measure. Consequently, their ground-truth forward representations are invariant modulo the kernel of the backward representation, i.e., $F^\star(s, a, z_{r'}) - F^\star(s, a, z_r) \in \ker(B^{\star T})$. Additionally, under the exact-FB rank conditions of Proposition 1,*

- *When $d = |\mathcal{S} \times \mathcal{A}|$, the kernel is trivial and $F^\star$ is fully invariant, i.e., $F^\star(s, a, z_{r'}) = F^\star(s, a, z_r)$.*
- *When $d > |\mathcal{S} \times \mathcal{A}|$, only the weaker statement holds and the corresponding necessary condition on Q-values is $Q^\star_{r'} = \nu Q^\star_r + \xi$.*

Proof. Let $n = |\mathcal{S} \times \mathcal{A}|$, $B^\star \in \mathbb{R}^{d \times n}$ be the matrix form of backward representations, and $r, \rho \in \mathbb{R}^n$ denote the reward and marginal-distribution vectors. For the transformed reward $r' = \nu r + \xi$ and r, Definition 2 gives us the corresponding latent variables

$$z_r = B^\star(r \odot \rho), \quad z_{r'} = B^\star(r' \odot \rho)$$

Expanding r', gives us

$$z_{r'} = \nu B^\star(r \odot \rho) + \xi B^\star(1 \odot \rho) = \nu z_r + \xi z_{\text{one}}$$

where $z_{\text{one}} := B^\star(1 \odot \rho)$ is the latent variable associated with constant one reward. By Definition 2, z_r and $z_{r'}$ index optimal policies and optimal Q-values corresponding to r and r' respectively.

We now show that these optimal policies are the same. By Lemma 3 for any policy π, $Q^\pi_{r'}(s, a) = \nu Q^\pi_r(s, a) + \xi$. Therefore,

$$Q^\star_{r'}(s, a) = \sup_\pi Q^\pi_{r'}(s, a) = \nu \sup_\pi Q^\pi_r(s, a) + \xi = \nu Q^\star_r(s, a) + \xi$$

Now consider any two action $a_1, a_2 \in \mathcal{A}$,

$$Q^\star_{r'}(s, a_1) - Q^\star_{r'}(s, a_2) = \nu[Q^\star_r(s, a_1) - Q^\star_r(s, a_2)]$$

Since $\nu > 0$, the sign of the difference is unchanged by positive affine transformation of reward. Hence the ordering of actions under $Q^\star_r$ and $Q^\star_{r'}$ is identical. This implies,

$$\arg\max_a Q^\star_{r'}(s, a) = \arg\max_a Q^\star_r(s, a) \; \forall \; s \in \mathcal{S}$$

Under the same deterministic tie-breaking rule, this implies $\pi^\star_{r'} = \pi^\star_r$.

Because the environment dynamics and the two optimal policies are identical, they induce the same successor measure,

$$M^{\pi^\star_{r'}} = M^{\pi^\star_r}$$

Without loss of generality, consider a fixed state-action pair (s, a). By the definition of ground-truth FB, we have the following equality,

$$\begin{aligned}
&F^\star(s,a,z_{r'})^T B^\star(s_f,a_f) = F^\star(s,a,z_r)^T B^\star(s_f,a_f)\ \forall\ (s_r,a_f) \in \mathcal{S}\times\mathcal{A} \\
&\implies [F^\star(s,a,z_{r'}) - F^\star(s,a,z_r)]^T B^\star(s_f,a_f) = 0 \\
&\implies [F^\star(s,a,z_{r'}) - F^\star(s,a,z_r)]^T B^\star = 0 \text{ (By stacking } B^\star(s_f,a_f) \text{ as columns)} \\
&\implies F^\star(s,a,z_{r'}) - F^\star(s,a,z_r) \in \ker(B^{\star T})
\end{aligned}$$

Since (s, a) was arbitrarily chose, the above result holds for all state-action pairs. Finally, under the exact-FB rank conditions of Proposition 1, we have $\text{rank}(B^\star) = n$ and $d \geq n$. We will now consider the above result in this context by the following two cases

- If $d = n$, then $B^\star$ is square, full rank and invertible matrix. Therefore $B^{\star T}$ is also invertible, resulting in $\ker(B^{\star T}) = \{0\}$. Hence, $F^\star(s,a,z_{r'}) = F^\star(s,a,z_r)$.
- If $d > n$, by rank-nullity, $\dim \ker(B^{\star T}) = d - n > 0$. Therefore, exact equality of the forward representations is not guaranteed; only invariance modulo $\ker(B^{\star T})$ is guaranteed.

Independently, the optimal Q-values always satisfy, $Q^\star_{r'}(s,a) = \nu Q^\star_r(s,a) + \xi,\ \nu > 0$ and $\xi \in \mathbb{R}$. This gives us the full result. □

C.5 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 γ) 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

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

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.6 Understanding the Coupled Learning Dynamics of FB

This section formalizes and proves a negative result for the coupled FB Bellman target map.

We begin by providing a formal definition of the coupled Bellman target map.

Definition 3 (Coupled FB Target Map)**.** *Let $\mathcal{Z}\subseteq\mathbb{R}^d$ be a finite latent space and $n=|\mathcal{S}\times\mathcal{A}|$. Let functions $f:\mathcal{S}\times\mathcal{A}\times\mathcal{Z}\to\mathbb{R}^d$ and $b:\mathcal{S}\times\mathcal{A}\to\mathbb{R}^d$ be given, and define the corresponding greedy latent-conditioned policy $\pi(a\mid s,z)=\delta\left(a\mid\arg\max_{a\in\mathcal{A}}f(s,a,z)^\top z\right)$. Assume* $\arg\max$ *uses a deterministic tie-breaking rule. For each latent $z\in\mathcal{Z}$, current state-action pair (s,a), and future state-action pair (s_f,a_f), the coupled FB target map, $\mathcal{H}_{F,B,z}:\mathbb{R}^{d\times n}\times\mathbb{R}^{d\times n}\to\mathbb{R}^{n\times n}$, is defined by*

$$[\mathcal{H}(f,b)]_z(s,a,s_f,a_f)\triangleq(1-\gamma)\frac{\delta(s_f,a_f\mid s,a)}{\rho(s_f,a_f)}+\gamma\mathbb{E}_{p(s'|s,a),\pi(a'|s',z)}\big[f(s',a',z)^\top b(s_f,a_f)\big]$$

As the target map $\mathcal{H}$ is not itself a self-map on the representation pairs, we analyze a restricted unfactorized setting in which $\mathcal{H}$ reduces to a well-defined self-map (operator) $\mathcal{T}_r$. We show that $\mathcal{T}_r$ can be discontinuous at a greedy-policy switching boundary; therefore, it is not a contraction in general. Consequently, the Banach fixed-point theorem cannot be directly extended to coupled FB Bellman update. This result does not prove non-convergence of FB; it suggests we need alternative theoretical tools to prove convergence of the practical FB learning algorithm, taking into account the additional regression step as well for the complete update map.

We begin by recalling that we analyze the coupled FB Bellman target map in a finite state-action space with $n=|\mathcal{S}\times\mathcal{A}|$; we assume we are given a full-support marginal distribution $\rho\in\mathbb{R}^n$ as a vector and $\text{diag}(\rho)\in\mathbb{R}^{n\times n}$ is a diagonal matrix. For a policy π, the policy-induced state-action transition matrix $P^\pi\in\mathbb{R}^{n\times n}$ is defined as $P^\pi((s,a),(s',a'))=p(s'|s,a)\pi(a'|s')$ and P^π is row-stochastic, i.e., $P^\pi.\mathbb{1}=\mathbb{1}$, where $\mathbb{1}\in\mathbb{R}^n$ is a one vector.

We consider the restricted setting from Sec. 3.3 where $d=n, B=I_n, z=\text{diag}(\rho)r$ with $r\in\mathbb{R}^n$ is a fixed reward vector. Under the notation that successor-measure ratio per latent vector is $Y_z=F_z^\top B$, this restriction gives us $Y:=Y_z=F_z^\top$. The reward-specific action-value is estimated as $Q_Y=Y\text{diag}(\rho)r$. Under the fixed deterministic tie-breaking, the induced greedy policy is given by $\pi_Y(s)=\arg\max_{a\in\mathcal{A}}Q_Y(s,a)$. We now define the space of normalized successor-measure ratio

matrices with non-negative entries as

$$\mathcal{X} = \{Y \in \mathbb{R}^{n\times n}_{\geq 0} : Y\rho = \mathbb{1}\}$$

They can be intermediate estimates and are not required to be the exact successor measure ratios induced by the optimal policies. We also recall that the entrywise supremum norm is defined as $\|Y\|_\infty := \max_{1\leq i,j\leq n}|Y_{ij}|$.

Definition 4 (Restricted Coupled Bellman Operator)**.** *For every $Y \in \mathcal{X}$, the restricted coupled Bellman operator is defined as*

$$\mathcal{T}_r(Y) = (1-\gamma)\mathit{diag}(\rho)^{-1} + \gamma P^{\pi_Y} Y$$

We now want to check that this is a well-defined self-map.

- The set $\mathcal{X}$ is closed in $(\mathbb{R}^{n\times n}, \|\cdot\|_\infty)$ and is therefore complete.
- $\mathcal{T}_r(Y)$ is non-negative.
- $\mathcal{T}_r(Y)\rho = (1-\gamma)\text{diag}(\rho)^{-1}\rho + \rho P^{\pi_Y} Y\rho = \mathbb{1}$.
- Therefore, $\mathcal{T}_r : \mathcal{X} \to \mathcal{X}$ is a well-defined self-map on a complete metric space.

Proposition 3. *In the restricted setting where the coupled Bellman target map $\mathcal{H}_{F,B,z}$ reduces to the self-map $\mathcal{T}_r : \mathcal{X} \to \mathcal{X}$, for every $\gamma \in (0,1)$, there exists a finite CMP with a full support marginal distribution ρ and a reward r such that $\mathcal{T}_r$ is discontinuous under the entrywise supremum norm. Therefore, $\mathcal{T}_r$ is not, in general, a γ-contraction under the entrywise supremum norm. Thus, the Banach fixed-point theorem is not applicable to the coupled FB target update.*

Proof. Consider a CMP with one state and three actions a_1, a_2, a_3. Every action transition deterministically results in the same state. Let $\rho = \frac{1}{3}I$ and $r = 3e_1 = 3 \odot [1,0,0]^\top$. This means $\text{diag}(\rho)r = e_1$.

For $0 < \epsilon < 0.1$, we define two candidate Y matrices as

$$Y^\epsilon_+ = \begin{bmatrix} 1+\epsilon & 1.9 & 0.1-\epsilon \\ 1 & 0.1 & 1.9 \\ 0.9 & 1 & 1.1 \end{bmatrix},\ Y^\epsilon_- = \begin{bmatrix} 1-\epsilon & 1.9 & 0.1+\epsilon \\ 1 & 0.1 & 1.9 \\ 0.9 & 1 & 1.1 \end{bmatrix}$$

Note that both matrices are non-negative and every row sums to 3. Therefore, $Y^\epsilon_+\rho = \mathbb{1}$, $Y^\epsilon_-\rho = \mathbb{1}$, and $Y^\epsilon_+, Y^\epsilon_- \in \mathcal{X}$ by the definition of $\mathcal{X}$.

The distance defined by entrywise supremum norm between these two candidate matrices is

$$\|Y^\epsilon_+ - Y^\epsilon_-\|_\infty = 2\epsilon$$

We next calculate the action-value estimates of the two candidate matrices as

$$Q_{Y^\epsilon_+} = \begin{bmatrix} 1+\epsilon \\ 1 \\ 0.9 \end{bmatrix},\ Q_{Y^\epsilon_-} = \begin{bmatrix} 1-\epsilon \\ 1 \\ 0.9 \end{bmatrix}$$

Since, $\pi_G = \arg\max_{a\in\mathcal{A}} Q_G(s,a)$, we get,

$$\pi_{Q_{Y^\epsilon_+}}(s) = a_1\ ,\ \pi_{Q_{Y^\epsilon_-}}(s) = a_2$$

As every action deterministically results in a transition to the same state s, applying $P^{\pi_{Q_{Y^\epsilon_+}}}$ to Y^ϵ_+ repeats the first row of Y^ϵ_+, and similarly, the second row of Y^ϵ_-.

Next, we want to compare $\|\mathcal{T}_r(Y^\epsilon_+) - \mathcal{T}_r(Y^\epsilon_-)\|_\infty$ and $\|Y^\epsilon_+ - Y^\epsilon_-\|_\infty$, and to conclude that $\mathcal{T}_r$ is not, in general, a γ-contraction.

1.

$$\begin{aligned}\|\mathcal{T}_r(Y^\epsilon_+) - \mathcal{T}_r(Y^\epsilon_-)\|_\infty &= \gamma\|P^{\pi_{Q_{Y^\epsilon_+}}} Y^\epsilon_+ - P^{\pi_{Q_{Y^\epsilon_-}}} Y^\epsilon_-\|_\infty \\ &= \gamma \max|[1+\epsilon, 1.9, 0.1-\epsilon] - [1, 0.1, 1.9]| \\ &= \gamma(1.8+\epsilon) \to \gamma 1.8 \text{ as } \epsilon \to 0\end{aligned}$$

2.
$$\|Y_+^\epsilon - Y_-^\epsilon\|_\infty = 2\epsilon \to 0 \text{ as } \epsilon \to 0$$

3.
$$\frac{\|\mathcal{T}_r(Y_+^\epsilon) - \mathcal{T}_r(Y_-^\epsilon)\|_\infty}{\|Y_+^\epsilon - Y_-^\epsilon\|_\infty} = \frac{\gamma(1.8+\epsilon)}{2\epsilon} \to \infty \text{ as } \epsilon \to 0$$

4.
$$Y_+^\epsilon = Y_-^\epsilon = Y^0 = \begin{bmatrix} 1 & 1.9 & 0.1 \\ 1 & 0.1 & 1.9 \\ 0.9 & 1 & 1.1 \end{bmatrix}$$

These calculations prove that $\mathcal{T}_r$ is discontinuous at Y^0 because if $\mathcal{T}_r$ were continuous, its distance should converge to zero as both images converge to $\mathcal{T}_r(Y^0)$. Therefore, $\mathcal{T}_r$ is not a contraction on $\mathcal{X}$. □

This proof uses intermediate candidate matrices rather than exact successor measure ratios. This is intentional for the contraction analysis: a contraction inequality must hold for every pair of elements in the operator's domain, including the intermediate candidate matrices during iterative estimation and not only the exact fixed point.

Remark 2. *At Y^0, actions a_1 and a_2 attain the same maximal action-value for the fixed reward $r = 3e_1$. The fixed deterministic tie-breaking rule selects one of the two actions, so π_{Y^0} and $\mathcal{T}_r(Y^0)$ are well-defined. If the rule selects a_1, discontinuity is witnessed by G_-^ϵ and similarly for a_2, discontinuity is witnessed at G_+^ϵ. So, $\mathcal{T}_r(Y^0)$ is discontinuous regardless of the chosen tie-breaking rule. Additionally, note that for every $\epsilon > 0$, both G_+^ϵ and G_-^ϵ have unique greedy actions. The tie occurs only at their common limiting matrix Y^0.*

In the remaining part of this section, we investigate whether a fixed behavior policy can result in a γ-contraction.

Let π_β be a fixed behavior policy and the self-map is defined accordingly as

$$\mathcal{T}^\beta(Y) = (1-\gamma)\text{diag}(\rho)^{-1} + \gamma P^{\pi_\beta} Y$$

Similar to $\mathcal{T}_r$, $\mathcal{T}^\beta$ preserves non-negativity and satisfies $\mathcal{T}^\beta(G)\rho = \mathbb{1}$ for every $Y \in \mathcal{X}$ and is therefore a valid self-map on $\mathcal{X}$.

For any $G, H \in \mathcal{X}$,

$$\begin{aligned} \|\mathcal{T}^\beta(G) - \mathcal{T}^\beta(H)\|_\infty &= \gamma \|P^{\pi_\beta}(G-H)\|_\infty \\ &\leq \gamma \|G - H\|_\infty \end{aligned}$$

This inequality follows from row-stochastic property of P^{π_β}. Therefore, $\mathcal{T}^\beta$ is a γ-contraction on the complete metric space of $\mathcal{X}$. By Banach fixed-point theorem, it has the unique fixed point

$$Y^\beta = (1-\gamma)(I_n - \gamma P^{\pi_\beta})^{-1}\text{diag}(\rho)^{-1}$$

Next, we conclude by illustrating the contraction in the same one-state CMP used in Proposition 3. Consider the behavior policy $\pi_\beta(a_i) = \beta_i$ such that $\beta_i \geq 0$ and $\beta_1 + \beta_2 + \beta_3 = 1$.

1.
$$P^{\pi_\beta} = \begin{bmatrix} \beta_1 & \beta_2 & \beta_3 \\ \beta_1 & \beta_2 & \beta_3 \\ \beta_1 & \beta_2 & \beta_3 \end{bmatrix}$$

2.
$$Y_+^\epsilon - Y_-^\epsilon = \begin{bmatrix} 2\epsilon & 0 & -2\epsilon \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

3.

$$\|\mathcal{T}^\beta(Y_+^\epsilon) - \mathcal{T}^\beta(Y_-^\epsilon)\|_\infty = 2\gamma\beta_1\epsilon \leq \gamma\|Y_+^\epsilon - Y_-^\epsilon\|_\infty$$

Thus $\mathcal{T}^\beta$ is a γ-contraction under the entrywise supremum norm. Additionally, this result highlights that the non-contraction result in Proposition 3 is caused by the coupling of the greedy policy and the current candidate Y.

Remark 3. *Proposition 3 established that the restricted coupled Bellman operator is not a γ-contraction in general. However, it does not conclude non-convergence or divergence of the operator. Additionally, since the complete representation update is a composition with a regression or factorization map, $\mathcal{R}_{fit}$, denoted as $\mathcal{U} = \mathcal{R}_{fit} \circ \mathcal{H}$, analysis of $\mathcal{U}$ (the complete update) may require analysis and assumptions on regression or factorization, optimization, sampling, target networks, and function approximation. We leave this for future work and have not discussed it here. Note also that $\mathcal{R}_{fit}$ is set-valued, and rotations, compensating scaling, and shear transformations leave the inner product $F^T B$ unchanged.*

C.7 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

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

Applying the SVD to the matrix $M^{\pi_\beta}\mathrm{diag}(\rho)^{-1}$, we have $M^{\pi_\beta}\mathrm{diag}(\rho)^{-1} = U_\beta\Sigma_\beta V_\beta^\top$, where U_β and V_β are two orthonormal matrices and Σ_β 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 4. *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 5. *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.8 One-Step FB Enables One Step of Policy Improvement

In the same way that the FB representations are defined in Definition 1, 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

$$F_\beta^\star(s,a)^\top B_\beta^\star(s_f,a_f) = \frac{M^{\pi_\beta}(s_f,a_f \mid s,a)}{\rho(s_f,a_f)}. \tag{24}$$

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^β using the behavioral backward representations as

$$z_r^\beta = \mathbb{E}_{(s_f,a_f)\sim\rho(s_f,a_f)}\left[B_\beta^\star(s_f,a_f)r(s_f,a_f)\right]. \tag{25}$$

Now, we have z_r^β indexing the behavioral Q-value $Q_r^\beta(s,a) = F_\beta^\star(s,a)^\top z_r^\beta$ (by Eq. 24 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 (Brandfonbrener et al., 2021; Eysenbach et al., 2022c; Park et al., 2025a;c) 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 Σ_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 (Σ_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 Loshchilov & Hutter (2019) 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 (Touati & Ollivier, 2021), 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 $\rho \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$[5], 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 analogue 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.

[5]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

$$\hat{\pi}(a \mid s, z) = \frac{\exp(\tau_{\text{policy}} F(s, a, z)^\top z)}{\sum_{a' \in \mathcal{A}} \exp\left(\tau_{\text{policy}} F(s, a', z)^\top z\right)}, \tag{26}$$

where τ_{policy} is a temperature for the softmax function fixed to $\tau_{\text{policy}} = 5 \times 10^{-3}$ during training. Following prior practice (Touati & Ollivier, 2021), 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, $Q_r^\star$, by running standard value iteration until convergence. We report the following four metrics as in Fig. 2 *(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_{one} is the latent variable corresponding to the all one reward vector. We sample ν and ξ randomly and compute the following MSE:

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

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 π_β. For simplicity, we set the policy π_β 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 π_β, 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:

$$\mathcal{L}_{\text{MC One-Step FB}}(F_\beta, B_\beta) = \|F_\beta B_\beta - M^{\pi_\beta}\text{diag}(\rho)^{-1}\|_F^2, \tag{27}$$

where the fixed successor measure M^{π_β} 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^{π_β} 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_β 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:

$$\epsilon_{Q^{\pi_\beta}} = \mathbb{E}_{z\sim D_{\text{eval}}}\left[\left\|\hat{Q}_r^{\pi_\beta}(z) - Q_r^{\pi_\beta}(z)\right\|_2^2\right] \tag{28}$$

- **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.8, 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 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:

$$\text{KL}_{\pi_{\text{one-step}}} = \frac{1}{|\mathcal{S}|}\mathbb{E}_{z\sim D_{\text{eval}}}\left[\sum_{s\in\mathcal{S}} D_{\text{KL}}\left(\pi_{\text{one-step}}(\cdot \mid s, z) \parallel \hat{\pi}_{\text{one-step}}(\cdot \mid s, z)\right)\right] \tag{29}$$

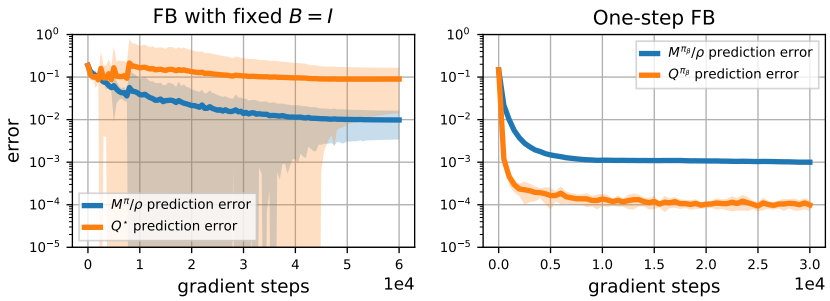


Figure 4: **Training curves for the Drunk Spider Gridworld didactic experiment (Fig. 3).** We train One-Step FB and FB with fixed $B = I$ for 3×10^4 and 6×10^4 gradient steps, by which point their training losses have plateaued.

- **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 ν and ξ randomly and compute the following MSE:

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

D.3 Drunk Spider Gridworld Didactic Experiment Details

In this section, we describe the environment, compared methods, and training procedure underlying the results in Sec. 5.2.

Environment. The environment is a deterministic 5×30 gridworld as seen in Fig. 3*(Left)*. The initial state s_0 is at (1,0) and the goal state G is at (1,29). Water occupies columns 2-27 and rows 0 and 2. The actions available are up, down, left and right. Actions that would leave the grid keep the agent in the same state. If the agent enters water or reaches the goal, the episode terminates. The reward is provided only at inference and is 1 on entering the goal state G and is 0 everywhere else. The optimal route is the short route with a discount factor of 0.99 and optimal value of 0.7547 in 29 steps. The lower best route takes 33 steps and has a value of 0.7250.

Data Collection. For this experiment, we consider an offline setting in which each method trains on a fixed dataset collected before representation learning. Each episode begins from a state randomly sampled uniformly from the 97 states on the grid excluding goal and water states. At each state, the behavior policy selects the first action along the shortest path from the current state to the goal state with probability 0.6 and randomly selects an action with probability 0.4. Episodes terminate upon reaching water, reaching the goal, or completing 64 environment steps. Each seed contains 100,000 reward-free transition tuples. During learning, FB and One-Step FB do not receive the reward function or the goal state G.

Methods Compared. We compare four methods.

1. **Value Iteration:** Value Iteration computes the optimal policy using the true transition and reward function using the Bellman Equation (Sutton et al., 1998; Agarwal et al., 2019).

$$Q_{k+1}(s,a) = \sum_{s'} P(s' \mid s,a)\left[r(s,a,s') + \gamma \max_{a'} Q_k(s',a')\right]$$

We initial Q-values with zeros and iterate until

$$\|Q_{k+1} - Q_k\|_\infty < 10^{-13}$$

The resulting policy, $\pi^\star(s) \in \arg\max_a Q^\star(s, a)$, follows the 29-step middle route, and achieves $V^\star(s_0) = 0.7547$.

2. **Behavioural occupancy measure with one-step of policy improvement ($V^{\textbf{greedy}(Q^{\pi_\beta})}$):** For each fixed offline dataset, empirical transition model, and behavior policy, we compute the behavioral successor measure using Eq. 1 for the fixed policy π_β. After the goal reward is revealed, we compute $Q_r^{\pi_\beta} = M^{\pi_\beta} r$, and one-step greedy policy as $\pi \in \arg\max_a Q_r^{\pi_\beta}$.
3. **One-Step FB:** Following the same architecture as in Sec. 3.3, we compute factorized representations from empirical transition and behavior policy estimated from the collected offline dataset. At inference, taking into account the goal reward function, we calculate the latent variable and take a greedy action under the predicted Q-function.
4. **FB with fixed $B = I$:** Following the same architecture as in Sec. 3.3, we set backward representation B to a scaled identity matrix to simplify the learning problem. During training, we use a softmax policy with a temperature of 0.05. At inference, we follow Definition 2 to obtain the zero-shot FB policy and corresponding predicted value function.

Training Procedure. We train on 20 seeds, collecting a fixed offline dataset of 10^5 reward-free transitions per seed. For both FB and One-Step FB, we use latent dimension $d = 150$. One-Step FB is trained for 3×10^4 gradient steps with a learning rate of 10^{-4}. We train FB for 6×10^4 with cosine decay of the learning rate from 3×10^{-5} to 10^{-6}. Fig 4 shows their losses plateau within these training budgets.

Evaluation. After training, we infer the zero-shot policies for each method. We then report the trajectories generated by 10 evaluation runs per training seed in Fig. 3*(Left)* and report the mean and standard deviation of value at state s_0 for each method in Fig. 3*(Right)*.

D.4 Rationales of Selecting Prior Methods

We compare One-Step FB against 6 prior unsupervised pre-training methods. The most relevant prior work is FB (Touati & Ollivier, 2021), 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 (Fujimoto et al., 2018)) to maximize the intrinsic reward derived from those state representations. Among them, we mainly compare against three families of methods. First, some prior methods train state representations via consistency along single-step or multi-step samples from the successor measure: Laplacian (Wu et al., 2018) and BYOL-γ (Lawson et al., 2025). Second, we also compare against TD-JEPA (Bagatella et al., 2025), a temporal-difference variant of joint-embedding predictive architecture (Assran et al., 2023) for zero-shot RL. TD-JEPA learns representations with a forward multi-step predictor similar to BYOL-γ, with an additional backward multi-step predictor to extract success features (latent representations). This method is a recent baseline for zero-shot RL. Third, another line of established methods derived state representations for the successor measure using variants of the expectile regression adapted from Kostrikov et al. (2021): HILP (Park et al., 2024b) and ICVF (Ghosh et al., 2023).

D.5 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 (Yarats et al., 2022) benchmark and a set of 4 state-based robotic navigation and manipulation domains from the OGBench (Park et al., 2024a) benchmark (Fig. 5). 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 (Tassa et al., 2018). Following prior work (Park et al., 2024b), 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:

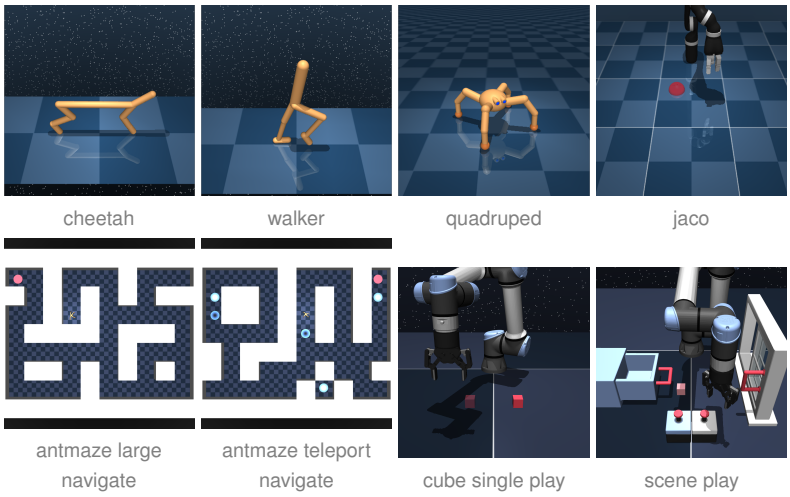


Figure 5: **Domains for evaluation.** *(Top)* ExORL domains (16 state-based tasks). *(Bottom)* OGBench domains (20 state-based tasks and 10 image-based tasks).

- `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. (2022), 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. (2022). 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 (Todorov et al., 2012) 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` do-

main 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. (2024a).

Datasets. On the ExORL benchmark, following the prior work (Touati et al., 2022; Park et al., 2024b; Kim et al., 2024; Zheng et al., 2025), we use 5×10^6 transitions collected by an exploration method (RND (Burda et al., 2019)) 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 be labeled with task-specific dense rewards, except in `jaco`, where the reward signals are sparse. See Yarats et al. (2022) for details of data collection.

On the OGBench benchmark, following the prior work (Zheng et al., 2025; Bagatella et al., 2025), 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 be labeled with *semi-sparse* rewards (Park et al., 2025b). See Park et al. (2024a) for details of data collection.

Evaluation Protocols. We compare the performance of One-Step FB against the 6 baselines (Sec. D.4) by pre-training each method for 10^6 gradient steps on different domains (5×10^5 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.5) 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 (Park et al., 2025b; Tarasov et al., 2023), we do *not* report the best performance during pre-training and instead report the evaluation results averaged over 8×10^5, 9×10^5, and 10^6 gradient steps for state-based domains. For image-based tasks, we report the evaluation results averaged over 4×10^5, 4.5×10^5, and 5×10^5 gradient steps. Following prior work (Park et al., 2025b), 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 (Fujimoto et al., 2018). 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.6 Implementations and Hyperparameters

We compare One-Step FB against 6 baselines, measuring the performance of undiscounted cumulative returns and success rates on downstream tasks. We implement One-Step FB and all baselines using JAX (Bradbury et al., 2018), adapting the OGBench (Park et al., 2024a) codebase. 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 (Tirinzoni et al., 2025; Touati et al., 2022), 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_β, the backward representation B_β, 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 (Park et al., 2024a), 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 (Tirinzoni et al., 2025), before inferring the latent variable z_r^β using zero-shot transitions (Eq. 25), we apply softmax weights based on the rewards of these transitions $\{(s_i, a_i, r_i)\}_{i=1}^N$, resulting in the following new rewards:

$$\tilde{r}(s_i, a_i) = w_i \cdot r(s_i, a_i), w_i = \frac{\exp\left(\tau_{\text{reward}} \cdot r(s_i, a_i)\right)}{\sum_{j=1}^N \exp\left(\tau_{\text{reward}} \cdot r(s_j, a_j)\right)}, \tag{30}$$

where τ_{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 λ_{BC}, the orthonormalization regularization coefficient λ_{ortho}, and the reward weighting temperature τ_{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 (Touati & Ollivier, 2021). Our FB implementation adapts the implementation from Tirinzoni et al. (2025) 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 (Barreto et al., 2017; Park et al., 2024b):

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

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 (Kostrikov et al., 2021). They differ in how they decompose the successor measure. For BYOL-γ and Laplacian, 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

Table 3: Common hyperparameters for One-Step FB and prior methods.

Hyperparameter	Value
optimizer	Adam (Kingma & Ba, 2015)
batch size	1024 on state-based domains, 256 on image-based domains
learning rate	1×10^{-4}
actor MLP hidden layer sizes	$(1024, 1024, 1024)$ on ExORL domains $(512, 512, 512, 512)$ on OGBench domains
value MLP hidden layer sizes	$(1024, 1024, 1024)$ on ExORL domains $(512, 512, 512, 512)$ on OGBench domains
representation MLP hidden layer sizes	Laplacian, BYOL-γ, ICVF, and HILP: $(256, 256, 256)$ on ExORL domains FB and One-Step FB: $(1024, 1024, 1024)$ for forward representations on ExORL domains, $(512, 512)$ for backward representations on ExORL domains All methods: $(512, 512, 512, 512)$ on OGBench domains
MLP layer normalization	No for all methods except TD-JEPA (see the open-source implementation)
MLP activation function	ReLU (Nair & Hinton, 2010) on ExORL domains GELU (Hendrycks & Gimpel, 2023) on OGBench domains
discount factor γ	0.98 on ExORL domains, 0.99 on OGBench domains
target networks update coefficient	1×10^{-2} on ExORL domains, 5×10^{-3} on OGBench domains
representation dimension d	50 on ExORL domains and 128 on OGBench domains For representation dimension of TD-JEPA, see Table 4
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 (Espeholt et al., 2018; Park et al., 2025b)
image augmentation method	random cropping
image augmentation probability	0.5
image frame stack	3

Table 4: **Domain-specific hyperparameters for One-Step FB and prior methods.** Following prior work Park et al. (2025b), 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.6.

	Laplacian		BYOL-γ		TD-JEPA			ICVF		HILP		FB			One-Step FB		
Domain	λ_{BC}	λ_{ortho}	λ_{BC}	λ_{ortho}	λ_{BC}	λ_{ortho}	d	λ_{BC}	μ	λ_{BC}	μ	λ_{BC}	λ_{ortho}	τ_{reward}	λ_{BC}	λ_{ortho}	τ_{reward}
`walker`	30	1	10	0.01	0	0	50	0.03	0.5	0.3	0.9	0	0.03	10	0	0.1	10
`cheetah`	10	1	3	0.01	0.3	1.0	50	0.3	0.5	3	0.5	0	0.3	10	0	1	3
`quadruped`	10	1	3	0.01	0	0	50	0.03	0.5	3	0.9	0	1	10	0	0.03	3
`jaco`	30	1	10	1	0	0.1	50	0.03	0.5	0.3	0.9	0	0.3	10	0	0.03	3
`antmaze large navigate`	1	0.1	3	0	0.03	0.1	128	3	0.5	1	0.5	0.03	0	10	0.03	0	10
`antmaze teleport navigate`	30	0	1	0	0.3	0.1	50	0.3	0.5	1	0.5	0.03	0	3	0.1	0	10
`cube single play`	1	0	30	0	0.3	0.1	50	30	0.5	1	0.5	0.3	0	10	0.3	0.3	300
`scene play`	1	0	3	0	3.0	0	50	0.3	0.5	1	0.9	0.3	0	3	0.3	0	300
`visual cube single play`	-	-	30	0	0.3	1.0	50	-	-	1	0.5	1	0	300	1	0	300
`visual scene play`	-	-	3	0	0.1	1.0	50	-	-	3	0.5	0.3	0	300	0.3	0	10

samples from the successor measure. After learning state representations, we use the same TD3 + BC (Fujimoto & Gu, 2021) 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 (Park et al., 2024b). The HILP implementation is adapted from the official implementation (Park et al., 2024b). 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 (Ghosh et al., 2023). 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.

BYOL-γ (Lawson et al., 2025). BYOL-γ 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-γ 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 (Wu et al., 2018). Unlike BYOL-γ, 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.

TD-JEPA (Bagatella et al., 2025). TD-JEPA is a latent-predictive zero-shot RL method that learns a state encoder $\phi : \mathcal{S} \to \mathbb{R}^{d_\phi}$, a task encoder $\psi : \mathcal{S} \to \mathbb{R}^{d_\psi}$, policy-conditioned predictors $T_\phi : \mathbb{R}^{d_\phi} \times \mathcal{A} \times \mathcal{Z} \to \mathbb{R}^{d_\psi}$ and $T_\psi : \mathbb{R}^{d_\psi} \times \mathcal{A} \times \mathcal{Z} \to \mathbb{R}^{d_\phi}$, and a latent-conditioned policy

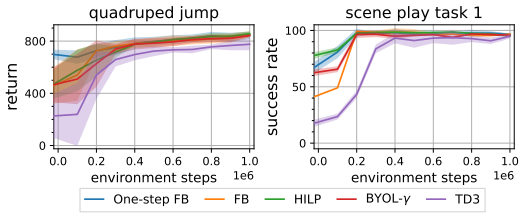


Figure 6: **Fine-tuning pre-trained agents on downstream tasks.** After offline pre-training, we conduct online fine-tuning on various methods using the same off-the-shelf RL algorithm (TD3). One-step FB continues to provide higher sample efficiency (+40% on average) during fine-tuning, as compared with the original FB method.

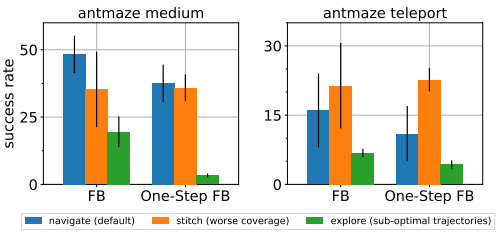


Figure 7: **The effects of the dataset quality on FB and One-Step FB.** Using `stitch` dataset has minor effects on both FB and One-Step FB, and sometimes even improves the performance, while using `explore` datasets reduces performance for both methods.

$\pi(a \mid \phi(s), z)$. Intuitively, the predictor T_ϕ estimates the successor features of the latent-conditioned policy π_z in the task-representation space:

$$T_\phi(\phi(s), a, z) \approx \mathbb{E}_{s_f \sim M^{\pi_z}(\cdot|s,a)} \left[\psi(s_f)\right].$$

TD-JEPA uses a temporal-difference latent-predictive loss to train the state encoder ϕ, task encoder ψ, and the policy-conditioned predictor T_ϕ:

$$\mathcal{L}_{\text{TD-JEPA}}(\phi, T_\phi, \psi) = \mathbb{E}_{\substack{(s,a,s') \sim \mathcal{D},\, z \sim p_{\mathcal{Z}}(z), \\ a' \sim \pi(\cdot|\bar{\phi}(s'), z)}} \left[\left\| T_\phi(\phi(s), a, z) - \bar{\psi}(s') - \gamma \bar{T}_\phi(\bar{\phi}(s'), a', z) \right\|_2^2 \right],$$

where $\bar{\phi}$, $\bar{\psi}$, and $\bar{T}_\phi$ denote target networks. Additionally, this method also trains the task encoder ψ through the symmetric objective $\mathcal{L}_{\text{TD-JEPA}}(\psi, T_\psi, \phi)$, obtained by swapping the roles of ϕ and ψ:

$$\mathcal{L}_{\text{TD-JEPA}}(\psi, T_\psi, \phi) = \mathbb{E}_{\substack{(s,a,s') \sim \mathcal{D},\, z \sim p_{\mathcal{Z}}(z), \\ a' \sim \pi(\cdot|\bar{\phi}(s'), z)}} \left[\left\| T_\psi(\psi(s), a, z) - \bar{\phi}(s') - \gamma \bar{T}_\psi(\bar{\psi}(s'), a', z) \right\|_2^2 \right].$$

Following TD-JEPA, we regularize the learned encoders with an orthonormality regularization similar to Eq 8 in One-Step FB (see Algorithm 1 in (Bagatella et al., 2025) for details). The representation objective is therefore

$$\mathcal{L}(\phi, T_\phi, \psi, T_\psi) = \mathcal{L}_{\text{TD-JEPA}}(\phi, T_\phi, \psi) + \mathcal{L}_{\text{TD-JEPA}}(\psi, T_\psi, \phi) + \lambda_{\text{ortho}} \left(\widehat{\mathcal{L}}_{\text{ortho}}(\phi) + \widehat{\mathcal{L}}_{\text{ortho}}(\psi) \right).$$

In our experiments, we tune the behavioral-cloning regularization coefficient $\lambda_{\text{BC}} \in \{0, 0.03, 0.1, 0.3, 1, 3\}$ and the orthonormalization regularization coefficient $\lambda_{\text{ortho}} \in \{0, 0.01, 0.1, 1.0\}$ to find the best values for different domains. We report the best selected hyperparameters in 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.5, 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 (Fujimoto et al., 2018) agent for fine-tuning. We selectively compare One-Step FB against 3 baselines in our offline experiments: BYOL-γ, HILP, and FB, and also include the performance of a TD3 agent trained from scratch for reference.

As shown in Fig. 6, 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.

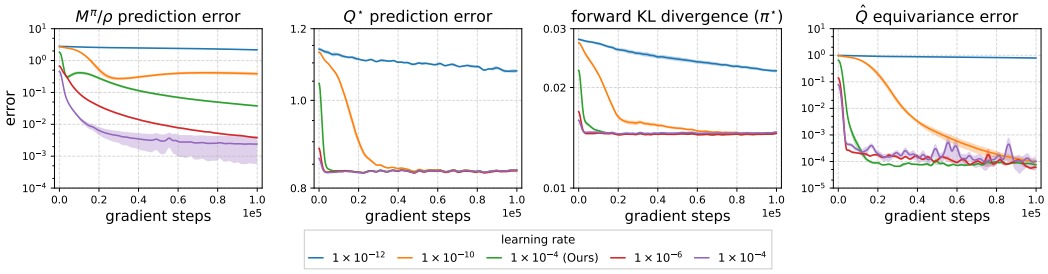


Figure 8: **Learning rate ablations for FB on the three-state CMP.** We conduct ablations to study the effect of learning rate on FB. Using learning rates other than 10^{-4} results in higher equivariance error of Q-value prediction ($\epsilon_{\text{equiv}} > 10^{-4}$). Thus, the failure mode of FB 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 (Nakamoto et al., 2023). In particular, we do not retain the pre-training data when performing online fine-tuning (Zhou et al., 2025), 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 suboptimal 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. 7 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 suboptimal 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 in Sec. 5.1 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. As seen in Fig. 8, 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 τ_{policy} affect the convergence of FB?* As mentioned in Appendix D.1, we use a softmax policy (Eq. 26) with temperature τ_{policy} to approximate the greedy

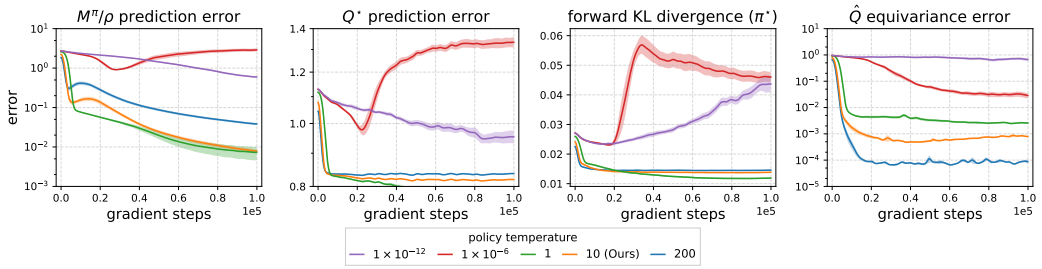


Figure 9: **Policy temperature ablations for FB on the three-state CMP.** We study the effect of policy temperature τ_{policy} on the convergence of FB: a decreasing τ_{policy} results in an increasing Q prediction equivariance error ϵ_{equiv}, suggesting that FB still fails to converge to the ground-truth fixed point.

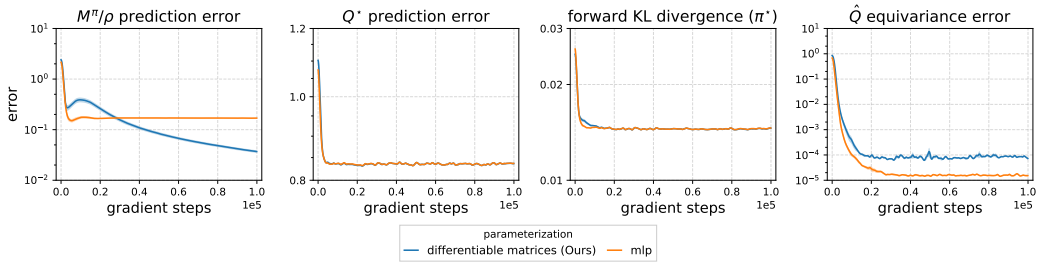


Figure 11: **Representation parameterization ablations for FB on the three-state CMP.** We study the effect of representation parameterization on the convergence of FB: using differentiable matrices yields lower error in successor measure ratio predictions, while using MLP parameterizations results in lower equivalence errors in the Q-value prediction. However, the similar failure mode for FB's convergence persists.

policy with respect to the inner product $F(s, a, z)^\top z$. One confounding factor is the choice of the temperature τ_{policy}. We conduct ablation experiments studying the effects of the policy temperature τ_{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. 9. We observe that a decreasing τ_{policy} results in an increasing Q prediction equivariance error ϵ_{equiv}, suggesting that τ_{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. 11 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.

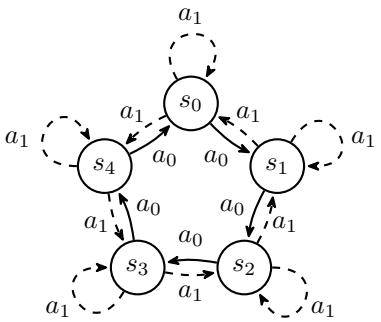


Figure 10: **The five-state circular CMP.** Agents start from state s_0 and take action a_i ($i = 0, 1$). At every state s_i, choosing a_0 convergence transits to the next state $s_{(i+1) \bmod 5}$, forming circular transitions. At every state s_i, choosing action a_1 transits to state $s_{(i-1) \bmod 5}$ 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.

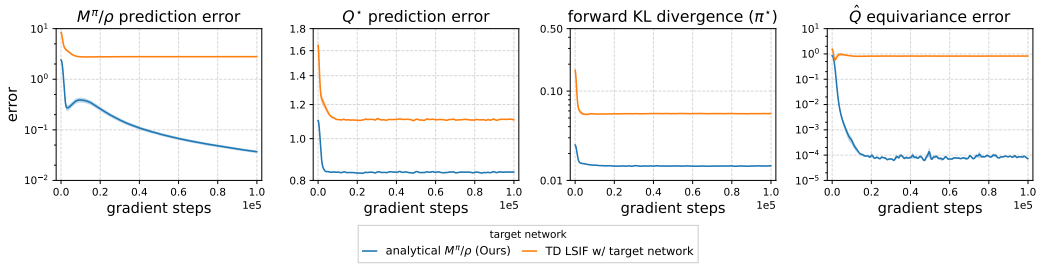


Figure 12: **Representation learning objective ablations for FB on the three-state CMP.** We conduct ablations to compare two representation learning objectives in our didactic experiments: *(1)* analytically compute the successor measure ratio and conduct bilinear decomposition into FB representations, 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.

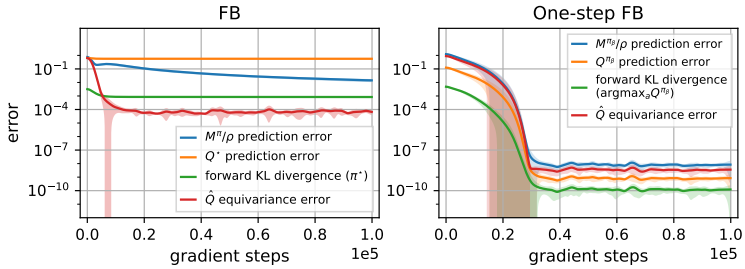


Figure 13: **Learning FB representations in the five-state circular CMP.** *(Left)* After training for 10^5 gradient steps, FB fails to converge to a pair of ground-truth FB representations. *(Right)* Given a fixed policy, One-Step FB exactly fits the ground-truth One-Step FB representations within 4×10^4 gradient steps. These observations are consistent with our analysis on the three-state CMP (Sec. 5.1).

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 space, we choose to compute the successor measure ratio $M_{\mathcal{Z}}^{\pi}\text{diag}(\rho)^{-1}$ 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. 12) 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. Specifically, we construct a five-state circular CMP (Fig. 10), 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. 13, we track the important statistics for both FB and One-Step FB, similar to Sec. 5.1. 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 4×10^4 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.6, there are four key hyperparameters of One-Step FB: *(1)* the behavioral-cloning regularization coefficient λ_{BC}, *(2)* the orthonormalization regularization coefficient λ_{ortho}, *(3)* the reward weighting temperature τ_{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 λ_{BC}, ablating over different values of λ_{BC} within $\{0, 0.03, 0.3, 3, 30\}$. We selectively show the comparisons between several λ_{BC} values in Fig. 14a. 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 (Park et al., 2024b), where they also exclude the behavioral-cloning regularization on ExORL tasks. In contrast, on OGBench domains, selecting a non-zero λ_{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 λ_{ortho} in Fig. 14b. Overall, the performance of One-Step FB is sensitive to the choice of λ_{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 λ_{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 λ_{ortho} is key to the performance of One-Step FB.

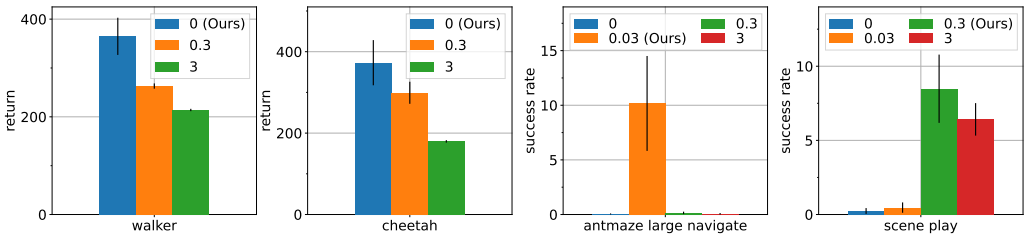
We also study the effect of our reward weighting strategy in Eq. 30 by ablating over different temperatures, $\tau_{\text{reward}} \in \{3, 10, 30, 300\}$, in the softmax function. In Fig. 14c, we compare the zero-shot performance of three variants of One-Step FB on each domain. These results suggest that One-Step FB is less sensitive to the choice of τ_{reward} on each domain. However, using larger values of τ_{reward} can slightly boost performance on OGBench domains. Therefore, we still tune the reward weighting temperature τ_{reward} for each domain separately and select the best candidates.

Finally, we study the effects of the representation dimension d. Both prior work (Park et al., 2024b; Touati et al., 2022) 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. 14d, 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. (2024b) and Touati et al. (2022), 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 (Bagatella et al., 2025), although $d = 128$ gives better performance on some domains.

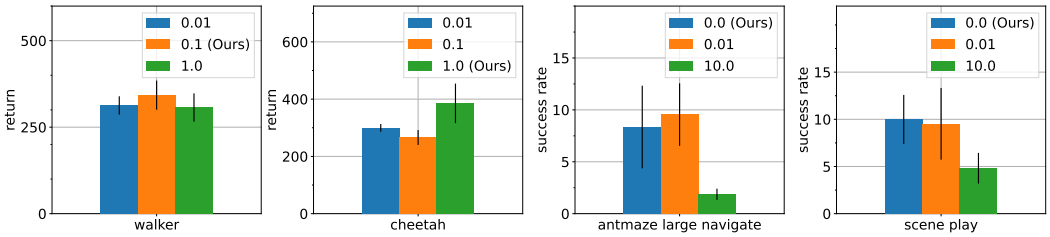
Taken together, we tune the behavioral-cloning regularization coefficient λ_{BC}, the orthonormalization coefficient λ_{ortho}, the reward weighting temperature τ_{reward}, and the representation dimension d on different domains. In general, our choices of hyperparameters are effective for One-Step FB on both ExORL and OGBench domains.

E.5 Comparison of One-Step FB With TD-JEPA In The Released TD-JEPA Codebase

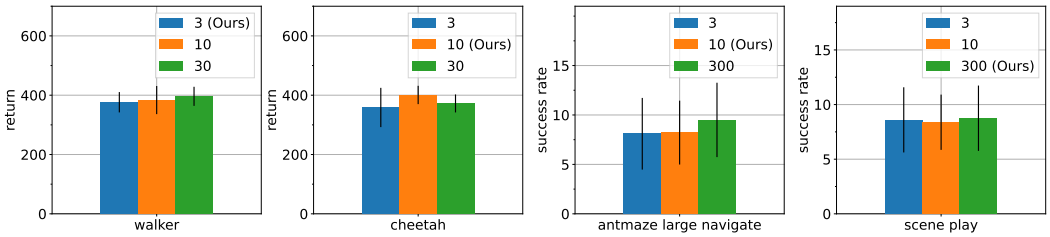
In this section, we rerun TD-JEPA as per the released TD-JEPA codebase (Bagatella et al., 2025). We then implement One-Step FB (Ours) in this codebase to provide a fair comparison. We first report results followed by the One-Step Hyperparameters per domain. The experiments were



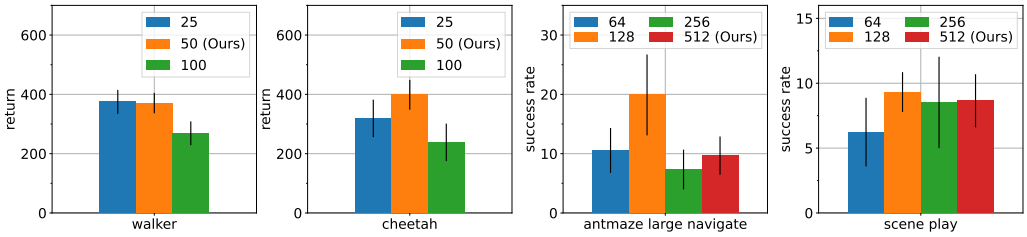
(a) Behavioral-cloning regularization coefficient λ_{BC}.



(b) Orthonormalization regularization coefficient λ_{ortho}.



(c) Reward weighting temperature τ_{reward}.



(d) Representation dimension d.

Figure 14: **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`.

conducted on five state-based OGBench domains: `antmaze medium navigate`, `antmaze large navigate`, `scene play`, `puzzle-3 x 3 play`, and `cube single play`.

E.5.1 Results

Table 5 compares published results from TD-JEPA, our rerun of TD-JEPA, and One-Step FB with the best hyperparameters. Our rerun reproduces the published TD-JEPA results. Across the five domains, relative to TD-JEPA's reported FB results, One-Step FB achieved $1.17\times$ the average success rate. One-Step FB exceeds FB on three out of 5 domains while remaining close on `antmaze medium navigate`. This is consistent with the findings reported in Sec. 5.3.

Table 5: **Direct comparison with TD-JEPA on state-based OGBench domains.** We report success rate (%, mean ± standard error). Published results are from Table 1 of TD-JEPA (Bagatella et al., 2025) and use 10 seeds. Our TD-JEPA rerun and One-Step FB use the same 10 seeds. Bold denotes performance within 95% of the best mean in each column. * denotes representation-learning methods whose zero-shot variants were introduced in TD-JEPA. (+−) indicate the performance of One-Step FB relative to FB.

Method	antmaze medium navigate	antmaze large navigate	scene play	puzzle-3x3 play	cube single play
Laplacian	50.00 ± 4.94	21.60 ± 3.90	7.80 ± 1.28	2.80 ± 0.68	15.11 ± 1.49
ICVF*	$\mathbf{79.80 \pm 2.62}$	$\mathbf{58.40 \pm 1.90}$	$\mathbf{45.40 \pm 2.29}$	16.60 ± 0.73	20.40 ± 1.93
HILP	$\mathbf{83.60 \pm 2.63}$	52.60 ± 3.86	$\mathbf{43.80 \pm 1.90}$	2.80 ± 0.68	74.20 ± 3.53
RLDP	74.60 ± 4.15	36.40 ± 4.66	11.60 ± 1.57	14.60 ± 0.90	19.80 ± 2.41
BYOL*	58.40 ± 2.00	26.60 ± 3.03	15.40 ± 1.37	13.20 ± 1.91	22.00 ± 3.16
BYOL-γ*	51.40 ± 1.55	21.80 ± 3.57	14.40 ± 2.32	18.80 ± 0.44	$\mathbf{79.40 \pm 2.83}$
FB	73.00 ± 2.72	36.80 ± 4.28	12.80 ± 1.61	4.80 ± 0.68	49.60 ± 3.83
TD-JEPA (paper)	70.40 ± 3.72	$\mathbf{57.20 \pm 4.25}$	38.44 ± 1.37	15.60 ± 1.11	34.20 ± 2.88
TD-JEPA (our rerun)	72.2 ± 2.3	51.4 ± 4.2	38.2 ± 1.9	16.2 ± 1.9	29.6 ± 2.4
One-Step FB (ours)	71.4 ± 3.5 (-1.6)	46.4 ± 3.7 (+9.6)	37.2 ± 4.1 (+24.4)	$\mathbf{20.0 \pm 0.0}$ (+15.2)	31.8 ± 2.3 (-17.8)

Table 6: **Selected One-Step FB hyperparameters in the released TD-JEPA codebase (Bagatella et al., 2025)** Configurations are selected using two seeds and evaluated using ten seeds. The reward temperature τ_{reward} is used only for task inference. All unlisted settings inherit the FB defaults, including $d = 50$.

Domain	λ_{ortho}	λ_{BC}	LR(B)	τ_{reward}
antmaze-medium-navigate	10^3	3	10^{-4}	10
antmaze-large-navigate	10^4	3	10^{-4}	3
scene-play	10^3	3	10^{-5}	3
puzzle-3x3-play	10^3	10	10^{-5}	3
cube-single-play	10^4	30	10^{-5}	3

E.5.2 Hyperparameters for One-Step FB

In this section, we provide additional details on hyperparamter search window and the best hyperparameters used for One-Step FB in Table 5. We use the reported hyperparameters of TD-JEPA and do not tune them further.

Hyperparameter Search. We conduct a domain-specific sweep, using the two random seeds, over

- orthonormalization coefficient:$\lambda_{\text{ortho}} \in \{10, 10^2, 10^3, 10^4\}$
- behavior-cloning coefficient: $\lambda_{\text{BC}} \in \{0.1, 0.3, 0.9, 1, 3, 9, 10, 30, 100\}$
- latent dimension: $d \in \{25, 50, 100, 200, 400\}$
- policy softmax temperature: $\tau_{\text{reward}} \in \{0.1, 0.3, 1, 2, 3, 5, 10, 30\}$
- task inference estimator: {`expectation`, `least-squares`, `softmax`}

Best Hyperparameters.

The selected hyperparameters are presented in Table 6.