A Unified Framework for Zero-Shot Reinforcement Learning

Jacopo Di Ventura, Jan Felix Kleuker, Aske Plaat, Thomas Moerland
j.di.ventura@liacs.leidenuniv.nl

Leiden University

Abstract

Zero-shot reinforcement learning (RL) has emerged as a setting for developing general agents, capable of solving downstream tasks without additional training or planning at test-time. While conventional RL optimizes policies for fixed rewards, zero-shot RL requires learning representations that enable immediate adaptation to arbitrary reward functions. As the field matures, the growing diversity of approaches demands a foundational framework reconciling different perspectives under a common unifying structure. In this work, we introduce a formal, unified framework for zero-shot RL, allowing for rigorous comparisons across methods. We propose a taxonomy organizing the algorithmic landscape along two levels: representation, distinguishing between compositional and direct methods based on their exploitation of action-value function decompositions; and learning paradigm, differentiating between reward-free and pseudo reward-free training.

1 Introduction

The reinforcement learning (RL; Sutton & Barto 2020) objective is commonly defined as identifying a policy that maximizes the expected cumulative reward. While this paradigm has enabled remarkable progress across diverse domains (Berner et al., 2019; Mirhoseini et al., 2021; Mankowitz et al., 2023; Seo et al., 2024), it is limited by its dependence on a single reward function, hindering the ability to transfer to new objectives.

Unsupervised RL relaxes this dependence by enabling agents to acquire environment knowledge in a task-agnostic pre-training phase, allowing for efficient fine-tuning once an extrinsic reward is revealed. Zero-shot RL (Touati et al., 2023) pushes this paradigm to its limit: agents must generalize to tasks immediately after pre-training, without any task-specific fine-tuning, planning, or substantial computation. As a result, this setting requires learning sufficiently expressive representations such that near-optimal behaviors can be extracted without fine-tuning. This paradigm has emerged as a possible candidate to train the analogue of foundation models in RL, so-called behavioral foundation models (Tirinzoni et al., 2025); an area where RL has historically lagged behind, motivating further research in the field (Yang et al., 2023). In recent years, zero-shot RL has received increasing attention, resulting in a diverse set of proposed algorithms (Hansen et al., 2020; Liu & Abbeel, 2021; Touati et al., 2023; Frans et al., 2024; Agarwal et al., 2025b; Cetin et al., 2025; Sun et al., 2025). Yet, the field remains fragmented: while prior works have offered partial unifying views (Touati et al., 2023; Agarwal et al., 2025a), some approaches lie outside their scope.

In this work, we establish the first unified framework for zero-shot RL, providing necessary structure to this emerging field. We introduce a taxonomy showing how methods can be systematically grouped, highlighting shared principles and differences, which, together with a consistent notation across reviewed methods, helps navigate the zero-shot landscape. In addition, Appendix A provides a unifying lens through which existing error bounds for zero-shot methods can be understood,

decomposing the total error into three primary contributing factors. Taken together, this work formalizes the field of zero-shot RL, providing a principled foundation on which future advancements can be built.

2 Preliminaries

Formally, the reinforcement learning problem is modeled as a Markov Decision Process (MDP) (Stratonovich, 1960; Puterman, 1994), defined as the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, p, p_0, r, \gamma)$. Here, $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $p_0 \in \Delta(\mathcal{S})$ is the initial state distribution, $p(s'|s,a) \in \Delta(\mathcal{S})$ is the transition function, $r(s,a,s')$ is the reward function (see Appendix E for argument reduction details) and $\gamma \in [0,1)$ is the discount factor.

The behavior of an agent interacting with this MDP is defined by a policy $\pi \in \Pi$, that is, a mapping $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ assigning a distribution over actions to each state. The state-value function $V^\pi(s)$ denotes the expected discounted return starting from state s and following policy π, $V^\pi(s) = \mathbb{E}_{\pi,p}[\sum_{t=0}^{\infty} \gamma^t\, r(s_t, a_t)|s_0 = s]$, resulting in the RL objective $\pi^*(\cdot|s) = \arg\max_{\pi\in\Pi} V^\pi(s)$.

A core concept in unsupervised RL is that of occupancy measures. The successor representation (SR; Dayan 1993) is simplest formulation of this concept. Formally, in a discrete MDP, for a given policy π, the SR is defined as the expected discounted future occupancy of state s^+ starting from state s:

$$M^\pi(s, s^+) = \mathbb{E}_{\pi,p}\Big[\sum_{t=0}^{\infty} \gamma^t \mathbb{1}\{s_t = s^+\} \mid s_0 = s\Big]. \tag{1}$$

This representation allows for a linear decoupling of the MDP dynamics from the reward function, where the value function is recovered as $V^\pi(s) = \sum_{s^+\in\mathcal{S}} M^\pi(s, s^+) r(s^+)$, providing a theoretical basis for zero-shot evaluation of a policy under varying rewards. This decomposition similarly extends to the action-value function.

Zero-Shot RL In the zero-shot setting, we consider a family of MDPs, $\mathcal{M}^{\mathcal{R}} \equiv \{(\mathcal{S}, \mathcal{A}, p, p_0, r, \gamma) \,|\, r \in \mathcal{R}\}$, where $\mathcal{R}$ denotes a specified set of reward functions under consideration, which may represent the space of all possible rewards or just a subset of tasks or skills of interest (Barreto et al., 2017). Under this setting, training is decoupled from task-specific feedback: while the agent may observe rewards during training, these signals are arbitrary and non-informative of the downstream tasks. At inference, reward functions are drawn from a distribution of downstream tasks $\mathcal{D}^{\text{test}}$, which is unknown during training. The objective is to obtain optimal policies π_r^* for all $r \in \mathcal{D}^{\text{test}}$ without additional parameter optimization, planning (reasoning over state transitions to synthesize new behaviors), or substantial computation. Rather than imposing an arbitrary threshold on what constitutes "substantial" computation, we recognize that the degree to which a method qualifies as zero-shot exists on a spectrum (discussed further in Section 7). The objective for zero-shot RL can formally be expressed as:

$$\pi^* = \arg\max_{\pi_\cdot\in\Pi} \mathbb{E}_{r\sim\mathcal{D}^{\text{test}}}\left[\mathbb{E}_{p_0,\pi_r(\cdot|s),p}\left[\sum_{t=0}^{\infty} \gamma^t\, r(s_t, a_t, s_{t+1})\right]\right]. \tag{2}$$

3 Framework

Our taxonomy categorizes the space of zero-shot methods by identifying two primary decision nodes: representation and learning paradigm. At the first level, we distinguish between methods based on their representation, partitioned into direct and compositional approaches. The taxonomy further divides based on the learning paradigm: reward-free vs. pseudo reward-free. This hierarchical decomposition, illustrated in Figure 1, serves as the basis for our framework. The following section provides a breakdown and formal definitions of these design primitives.

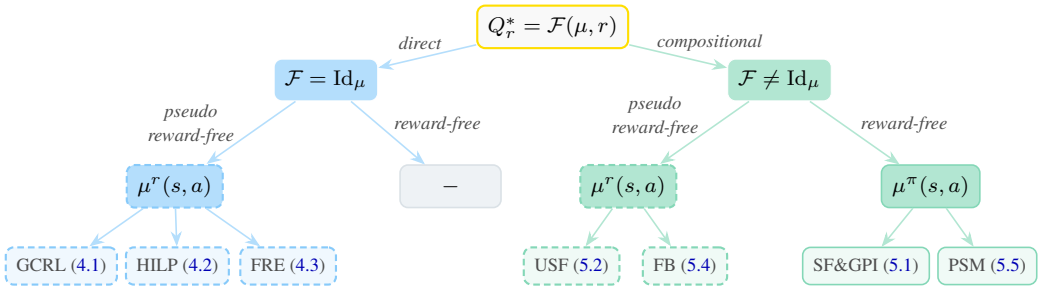

Figure 1: Visual taxonomy of the algorithmic landscape across two levels: **representation**, distinguishing direct and compositional methods based on their exploitation of value function decompositions; and **learning paradigm**, distinguishing reward-free from pseudo reward-free approaches.

3.1 Representation: Direct vs. Compositional

Direct methods provide a conceptually straightforward approach to zero-shot RL by learning a reward-conditioned value function $Q(s, a|r)$, also known as a universal value function (Schaul et al., 2015). In contrast, compositional methods decompose the value function by learning an intermediate target, enabling the reconstruction of task-specific value functions during inference.

Direct representations learn a direct mapping from state-action and reward to optimal values:

$$Q^* : \mathcal{S} \times \mathcal{A} \times \mathcal{R} \to \mathbb{R}, \quad (s, a, r) \mapsto Q_r^*(s, a). \tag{3}$$

Under this direct mapping, the function approximator must capture the complete reward-induced structure, yielding Q^* values directly from (s, a, r), with no explicit substructure between policy, occupancy, and value. Policy extraction for a target task $r \sim \mathcal{D}^{\text{test}}$ is performed via $\pi_r^*(s) \in \arg\max_a Q^*(s, a, r)$.

Compositional representations explicitly leverage the structure of the value function. Under this approach, the learning problem reduces to estimating individual components, that are recombined at inference time according to the specified substructure. More formally, compositional methods learn representations $\mu(s, a)$, which allow to infer optimal values via some structure in the form of a decomposition operator $\mathcal{F}$:

$$Q_r^*(s, a) = \mathcal{F}(\mu, r), \tag{4}$$

where the operator $\mathcal{F}$ encodes the relationship between the learned representation and the reward. An instantiation of $\mathcal{F}$ could be as simple as the inner product between some reward representation $f(r)$ and the target, i.e. $\mathcal{F}(\mu, r) = \langle \mu, f(r) \rangle$.

Notably, as shown in Figure 1, direct representations constitute the special case where $\mu^r = Q_r^*$, or equivalently $\mathcal{F} = \mathrm{Id}_\mu$. Because we define compositional methods to be everything that is not direct, we assume compositional methods to have a non-trivial decomposition operator $\mathcal{F} \neq \mathrm{Id}_\mu$.

3.2 Learning Paradigm: Reward-free vs. Pseudo reward-free

At test time, the agent is evaluated under an unknown reward distribution $\mathcal{D}^{\text{test}}$, to which it has no access during training. Consequently, it must either adopt a reward-free training paradigm or define an arbitrary set of reward functions to guide the learning process (pseudo reward-free).

Reward-free methods learn a quantity $\mu^\pi(s, a)$ – always superscribed by π – through objectives that are entirely independent of reward signals. By definition, this paradigm excludes any objective relying on reward maximization (i.e. Bellman optimality backups). Therefore, as illustrated in Figure 1, reward-free objectives only exist in the context of compositional representations.

As an example, consider a discrete-space MDP and the successor representation of a fixed policy π, denoted as M^π, learned by minimizing the TD-error of the SR following π. The representation

is compositional because it targets a component of the value function; as such, we denote it as $\mu^\pi = M^\pi$, with superscript π because it is learned via a reward-free objective. Zero-shot adaptation to any reward $r \in \mathcal{R}$, is performed via the operator $\mathcal{F}$ as $Q_r^\pi = \mathcal{F}(\mu^\pi, r) = \langle M^\pi, r \rangle$.

Pseudo reward-free methods, in contrast, leverage reward signals to learn representations contingent on a reward function, denoted by $\mu^r(s, a)$. The agent is provided access to a training distribution of random rewards, $r \sim \mathcal{D}^{\text{train}}$, which are non-informative of the downstream tasks, resulting in a from self-supervised RL. The underlying principle is that by sampling a sufficiently diverse set of random reward functions during training, the resulting representation space will cover the rewards encountered during inference.

Finally, we formalize the preceding concepts into a unified zero-shot RL framework.

Unified Zero-Shot RL Framework

Training
Define $\mu(s, a)$ and a compatible training paradigm such that

$$Q_r^* = \mathcal{F}(\mu, r) \quad \forall r \in \mathcal{R} \tag{5}$$

Policy Extraction
Given a (unseen) reward $r \sim \mathcal{D}^{\text{test}}$ extract a policy via

$$\pi_r^*(\cdot|s) = \arg\max_a \mathcal{F}(\mu, r) \tag{6}$$

In the following sections we review key zero-shot methods starting with direct methods in Section 4, and moving to compositional in Section 5. Appendix A, presents an analysis of existing error bounds under a common lens, categorizing the total error into three primary components providing insight into how design choices affect the error.

4 Direct Representations

Methods relying on direct representations (left branch in Figure 1) parameterize and optimize the value function directly. The training objective is the optimal value function itself, $\mu^r(s, a) = Q_r^*(s, a)$. While some methods instead learn a parameterized policy $\pi_r^*(s) \in \Delta(\mathcal{A})$, this often serves as a practical workaround for the intractability of $\arg\max_a Q_r^*(s, a)$ (Lillicrap et al., 2016; Fujimoto et al., 2018). Because we are only concerned with *optimal* behavior for each task, direct representations implicitly parameterize each policy by the reward function for which it is optimal. This approach is usually facilitated via an embedding function $f : \mathcal{R} \to \mathcal{Z}$ that maps reward functions into a latent representation space. This allows us to index the set of optimal policies by $\mathcal{Z}$, where π_z represents the optimal policy π_r^* for the reward function embedded as $z = f(r)$.

Defining a task space While f could theoretically be the identity mapping, this becomes intractable in continuous or high-dimensional settings. Consequently, training direct representations generally involves two phases: (i) define or learn a task encoder $f : \mathcal{R} \mapsto \mathcal{Z}$; (ii) train some representation of optimal action-value functions or policies $\mu^{f(r)}$ using rewards r sampled from some distribution until convergence. Since the second phase corresponds to standard RL training for each sampled reward, the primary challenge lies in the first phase: learning an encoder f that defines a task space $\mathcal{Z}$ expressive enough to retain all reward-relevant information while remaining sufficiently smooth to enable generalization through μ. Since the space of admissible rewards is vast, the effectiveness of generalization critically depends on how the chosen prior constrains and shapes coverage of plausible downstream tasks. In addition, task embeddings suffer from inherent identifiability issues: distinct reward functions may yield indistinguishable behaviors, making the corresponding task representations fundamentally ambiguous. In the following, we present three methods that successfully define or learn such a space.

4.1 Goal-Conditioned RL

A canonical example of such an embedding arises in goal-conditioned RL (GCRL; Schaul et al. 2015; Andrychowicz et al. 2017; Levy et al. 2019; Chebotar et al. 2021; Eysenbach et al. 2022; Park et al. 2023), where tasks are limited to goal reaching. In this setting, the reward embedding emerges naturally: each task is defined by a goal state, giving rise to the natural identification $\mathcal{Z} = \mathcal{S}$.

4.2 Hilbert Representations

The Hilbert Representations framework (HILP; Park et al. 2024) learns a geometric abstraction of the state space by training a mapping $\phi : \mathcal{S} \to \mathcal{Z}$, where $\mathcal{Z}$ is a Hilbert space, such that distances in the latent space correspond to temporal distances in the original MDP. Using this, we can define a goal-reaching reward function for any vector in latent space via $r_z(s, s') := (\phi(s') - \phi(s))^\top z$, rewarding transitions that align with the direction of the goal z. Through $r_z(s, s')$, optimal policies π_z^* can be trained for all $z \in \mathcal{Z}$. At test time, given a reward function r, the optimal behavior for any task can be recovered by $\pi_{f(r)}$ for $f(r) = \arg\min_{z \in \mathcal{Z}} [(r(s, a, s') - r_z(s, s'))^2]$.

4.3 Functional Reward Encodings

Restricting the task space to goal-reaching is not intrinsic to direct representations. The Functional Reward Encoding (FRE) framework (Frans et al., 2024) introduces a method to embed arbitrary reward functions into a latent space $f_e : \mathcal{R} \to \mathcal{Z}$. A state-dependent reward function $r : \mathcal{S} \to \mathbb{R}$ is represented by a set of samples $\{(s_i, r(s_i))\}_{i \in \mathcal{I}}$, which a permutation-invariant transformer encodes into a latent vector $z \in \mathcal{Z}$. To ensure z captures generalizable features of the reward landscape, an encoder-decoder architecture is trained using an information-bottleneck objective over a reward distribution $\mathcal{D}^{\text{train}}$ comprised of goal-reaching, linear, and random MLP functions. Then, networks $Q(\cdot, z)$, $V(\cdot, z)$, and $\pi(\cdot, z)$ are trained using Implicit Q-Learning (Kostrikov et al., 2022) under sampled reward functions projected into the latent space $\mathcal{Z}$ via the frozen encoder.

Relation to Framework For all direct methods, the training target is the optimal action-value function itself, i.e. $\mu = Q_r^*$. In general, μ cannot be conditioned directly on the reward function, as the reward space is generally intractable; instead, a suitable reward representation $f : \mathcal{R} \to \mathcal{Z}$ is required such that $\mu^{f(r)} = Q_r^*$. Given a reward function r, a policy can then be extracted as $\pi_r(\cdot|s) = \arg\max_a \mu^{f(r)}(s, a)$.

5 Compositional Representations

Central to compositional methods (right branch in Figure 1) is the concept of occupancy measures (such as the SR, Eq. 1), which serves as the foundation for many compositional methods (Agarwal et al., 2025a). Compositional methods can be both reward-free (Sections 5.1, 5.5) and pseudo reward-free (Sections 5.2, 5.4). In general, compositional methods aim to capture the occupancy measure of a set of policies, including degenerate case of a single policy, and extract relevant behavior given a reward. Depending on the specific method, this extraction may rely on an explicit search at test time or be shifted to pretraining by leveraging inductive biases.

This section presents both zero-shot methods and their foundational objects (Sections 5.1, 5.3), organized by conceptual evolution rather than training paradigm: starting with basic approaches, we trace how subsequent methods address specific limitations or explore distinct paths.

5.1 Successor Features

Successor Features (SF; Barreto et al. 2017) extend the SR to continuous MDPs by tracking expected discounted future *features* instead of states. Assume a feature mapping $\phi : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^d$ for some $d \in \mathbb{N}$, and suppose that the reward function can be linearly decomposed as

$$r(s, a, s') = \phi(s, a, s')^\top w, \tag{7}$$

where $w \in \mathbb{R}^d$ is a weight vector and $\phi(s, a, s') \in \mathbb{R}^d$ are the *basic features* of the one-step transition (s, a, s'). Note, that for a single reward function, such a decomposition always exists (e.g., $\phi(s, a, s') = r(s, a, s')$ and $w = 1$). Following Eq. (1), the successor features of a policy π are defined as

$$\psi^\pi(s, a) := \mathbb{E}_{\pi,p}\left[\sum_{t=0}^{\infty} \gamma^t \phi(s_t, a_t, s_{t+1}) \,\middle|\, s_0 = s, a_0 = a\right]. \tag{8}$$

Just as the SR decouples dynamics from rewards, SFs decouple dynamics from the reward weights w. Given a reward embedding w and basic features ϕ, the action-value function under policy π can be recovered as $Q_r^\pi(s, a) = \psi^\pi(s, a)^\top w$. This decomposition allows for zero-shot policy evaluation: the performance of π can be evaluated on any task in the span of ϕ by simply updating w.

Training Analogous to value functions, successor features satisfy a Bellman-like recursion:

$$\psi^\pi(s, a) = \mathbb{E}_{s' \sim p,\, a' \sim \pi}[\phi(s, a, s') + \gamma \psi^\pi(s', a')]. \tag{9}$$

Allowing ψ^π to be learned using standard temporal-difference (TD) methods.

While the linearity assumption of Eq. (7) may appear restrictive, the framework can still be applied when ϕ only *approximately* linearizes rewards by considering the best linear approximation $\tilde{w}_r$ such that $r \approx \phi^\top \tilde{w}_r$. For instance, $\tilde{w}_r$ can be obtained by minimizing the mean squared error:

$$\tilde{w}_r = \arg\min_w \mathbb{E}_{(s,a,s') \sim \mathcal{D}}\big[(r(s, a, s') - \phi(s, a, s')^\top w)^2\big]. \tag{10}$$

Generalized Policy Improvement Standard policy improvement (Sutton & Barto, 2020) identifies a better policy relative to the value function of a single policy, Generalized Policy Improvement (GPI) extends this logic to a set of policies $\{\pi_i\}_{i=1}^n$. Given the respective action-value functions $\{Q^{\pi_i}\}_{i=1}^n$ under the same reward, we define the GPI policy as the pointwise maximum over the set

$$\pi_{\text{GPI}}(\cdot|s) \in \arg\max_a \max_i Q^{\pi_i}(s, a). \tag{11}$$

Similarly to the standard policy improvement theorem, it can be shown the new policy guarantees that $Q^{\pi_{\text{GPI}}}(s, a) \geq \max_i Q^{\pi_i}(s, a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$ (Barreto et al., 2017).

Relation to Framework Pairing SFs with GPI is an instance of a reward-free approach, as the training target $\mu^\pi = \psi^\pi$ can be trained without any reward signal. From said target, the optimal value function can be recovered using the decomposition operator $\mathcal{F}(\mu^\pi, r) = \max_\pi (\mu^\pi)^T w_r$. In the policy extraction phase, given a reward function $r \sim \mathcal{D}^{\text{test}}$, we first calculate $\tilde{w}_r$ (Eq. 10), for which we can infer an approximation for the optimal policy via GPI

$$\pi_r(\cdot|s) = \arg\max_a \max_i \psi^{\pi_i}(s, a)^T \tilde{w}_r. \tag{12}$$

5.2 Universal Successor Features

The scalability of SF&GPI is primarily constrained by the limited generalization across policy space. One might mitigate this issue by conditioning the successor features on the policy itself (i.e., learning $\psi^\pi(s, a)$ as a function of π), yet a naive search over Π at test time remains computationally intensive without significant constraints. Universal Successor Features (USF; Borsa et al. 2019) circumvent this limitation by exploiting the relationship between optimal policies and their inducing reward functions. By parameterizing policies by the reward weights for which they are optimal, we avoid an explicit search over policies during the policy extraction phase. Specifically, for a reward function of the form $r = \phi^T w$, we define $\pi_w := \pi^*_{\phi^T w}$ as the optimal policy given the reward weights w. Then, the universal successor features for a family of policies $(\pi_w)_{w \in \mathcal{W}}$ with $w \in \mathbb{R}^d$ are defined as

$$\pi_w(\cdot|s) = \arg\max_a \psi(s, a, w)^\top w, \quad \psi(s, a, w) = \psi^{\pi_w}(s, a). \tag{13}$$

Notably, $\psi(s, a, w)$ encodes the successor features for the optimal policy with respect to all rewards admitting the factorization $r = \phi^\top w$. Therefore, the policy extraction phase for USF reduces to computing the task vector w as in Eq. (10).

Training As with successor features, USF also satisfy a Bellman-like equation, motivating minimization of the Bellman residual to train USFs. Following Eq. (9), we get:

$$L(\psi) = \mathbb{E}_{w \sim p_{\mathcal{W}}} \left[(\delta_w^t)^2 \right], \; \delta_w^t = \phi(s_t, a_t, s_{t+1}) + \gamma \, \psi(s_{t+1}, \pi_w(s_{t+1}), w) - \psi(s_t, a_t, w), \quad (14)$$

where δ_w^t is the temporal-difference error (Barreto et al., 2017). These objectives are particularly useful when evaluating policies on tasks different from those that induced them, as in GPI.

The curse of ϕ Successor Features rely on the assumption that all downstream tasks of interest lie in the linear span of the basic features ϕ (Eq. 7). Unless a unifying object encompassing both ψ and ϕ is introduced, they cannot be learned jointly. In such case, two distinct objectives must be defined: one for ψ and one for ϕ. Choosing an appropriate training objective for learning ϕ is critical, as it determines which aspects of the environment are preserved and, consequently, which reward functions admit accurate linear parameterizations. For a comprehensive review and comparison of the existing methods, we refer the reader to (Touati et al., 2023).

Relation to Framework The training target for USF methods is $\mu^r = \psi(s, a, w_r)$, allowing to recover the optimal values via the decomposition operator $\mathcal{F}(\mu^r, r) = (\mu^r)^\top w_r$. As such, we require some reward representations during training (cf. Eq. 14), rendering this a pseudo reward-free approach. During inference, when exposed to a reward r, we calculate $\tilde{w}_r$ through Eq. (10) and infer a policy through $\pi_r(\cdot|s) = \arg\max_a \psi(s, a, \tilde{w}_r)^\top \tilde{w}_r$.

5.3 Successor Measures

While feature-based methods adapt successor representations to continuous domains by tracking state properties, the successor measure (SM; Blier et al. 2021) provides a more fundamental generalization. By transitioning from discrete counts to a measure-theoretic framework, it extends the SR to continuous state-action spaces while preserving its core interpretation. We overload the notation by identifying the SM as M, just as with the SR, as the SM naturally encompasses the SR.

Formally, the successor measure for a policy π, a state–action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$, and a set $X \subset \mathcal{S}$ is defined as

$$M^\pi(s, a, X) := \mathbb{E}_{\pi, p} \left[\sum_{t=0}^{\infty} \gamma^t \, \mathbb{1}\{(s_t, a_t) \in X\} \,\middle|\, s_0 = s, a_0 = a \right]. \quad (15)$$

Treating M^π as a measure allows us to handle discrete and continuous spaces in a unified way. Let $\rho \in \Delta(\mathcal{S})$ be a state distribution satisfying $\rho(s) > 0$ for all $s \in \mathcal{S}$, and define the Radon–Nikodym derivative $m^\pi := \frac{\mathrm{d}M^\pi}{\mathrm{d}\rho}$, or in other words the density of M^π with respect to ρ. For $\gamma < 1$, M^π is a finite measure since $M^\pi(s, a, \mathcal{S}) = (1-\gamma)^{-1}$, which, together with the assumption on ρ, guarantees that this density exists. We can then express the action–value function under a reward $r : \mathcal{S} \to \mathbb{R}$ as $Q^\pi(s, a) = \mathbb{E}_{s^+ \sim \rho}[m^\pi(s, a, s^+) \, r(s^+)]$.

Training Successor measures follow a Bellman-like relationship, allowing $m_\theta \approx m^\pi$ to be trained via temporal-difference learning, i.e.:

$$L(\theta) = \mathbb{E}_{s, a \sim \rho, s^+ \sim \rho}[(m_\theta(s, a, s^+) - \hat{m}_{\bar{\theta}}(s, a, s^+))^2], \quad (16)$$

Where the target $\hat{m}_{\bar{\theta}}(s, a, s^+) = \frac{\delta_s(\mathrm{d}s^+)}{\rho(\mathrm{d}s^+)} + \gamma \mathbb{E}_{s' + \sim p(\cdot|s,a), a' \sim \pi(s')}[m_{\bar{\theta}}(s', a', s^+)]$. As in classical RL, the gradient of the loss is not propagated through the target, and $\bar{\theta}$ refers to a stop-gradient or target network. Notably, in the following subsections we only have to deal with the Radon-Nikodym density m^π, and assume that ρ is a fixed (data) distribution.

Relation to Framework While the SM is not a standalone framework in itself, in its most basic formulation it can describe a reward-free, compositional method with target $\mu^\pi = m^\pi$, and the optimal value function can be recovered via the decomposition operator

$$\mathcal{F}(\mu^\pi, r) = \max_{\pi \in \Pi} \mathbb{E}_{s^+ \sim \rho}[\mu^\pi(s, a, s^+)\, r(s^+)], \tag{17}$$

policy extraction is facilitated as $\pi_r(s) = \arg\max_a \max_{\pi \in \Pi} \mathbb{E}_{s^+ \sim \rho}[\mu^\pi(s, a, s^+)\, r(s^+)]$. Notably, if Π is sufficiently expressive, specifically if it contains the optimal policy π^* for a given reward r, then full state-action coverage (support) is a sufficient condition for recovery of the optimum. In contrast, if the search space is restricted to a single SM, encoding, for example, a behavior policy m^{π_b}, coverage alone does not guarantee optimality. Assuming m^{π_b} encodes the necessary transitions to define π_r^*, the policy extraction operation $\arg\max_a \mathbb{E}_{s^+ \sim \rho}[m^{\pi_b}(s, a, s^+) r(s^+)]$ lacks the ability to break the temporal structure induced by the behavior policy itself, making optimality guarantees a function of both temporal structure and coverage. Nonetheless, an approach following this framework, has shown good empirical performance (Zheng et al., 2026). Lastly, it should be noted that SF and SM are related as $\psi^\pi(s, a) = \mathbb{E}_{s^+ \sim \rho}[m^\pi(s, a, s^+)\phi(s^+)]$ (Touati et al., 2023).

5.4 Forward-Backward Representation

Approaches that directly approximate the successor measure have the advantage of tracking successor states rather than features, thus not requiring a secondary objective to learn the basic features. Forward-Backward representations (FB; Blier et al. 2021; Touati & Ollivier 2021; Touati et al. 2023) factorize the successor measure into a low-rank decomposition:

$$m^\pi(s, a, s^+) = F(s, a)^\top B(s^+), \tag{18}$$

where $F : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ and $B : \mathcal{S} \to \mathbb{R}^d$. As for the SM, reward-free objectives can be defined for FB, such as Eq. (16). However, naive policy-conditioned approaches may suffer from an expensive test-time search over the representation to extract the optimal policy.

Later works (Touati & Ollivier, 2021), propose to employ a similar strategy to that of USF to FB, namely, relating policy parametrization and rewards. For a parametric family of policies $(\pi_z)_{z \in \mathcal{Z}}$, we define the forward backward representation as as a pair of functions $F : \mathcal{S} \times \mathcal{A} \times \mathcal{Z} \to \mathcal{Z}$ and $B : \mathcal{S} \to \mathcal{Z}$ such that for all $z \in \mathcal{Z}, (s, a) \in \mathcal{S} \times \mathcal{A}$

$$\pi_z(\cdot|s) = \arg\max_a F(s, a, z)^\top z, \quad m^{\pi_z}(s, a, s^+) = F(s, a, z)^\top B(s^+). \tag{19}$$

Crucially, for any bounded reward function r, by defining z as the reward projection $z := \mathbb{E}_{s \sim \rho}[B(s) r(s)]$ we can recover the optimal value function as $Q_r^*(s, a) = F(s, a, z)^\top z$ (Theorem 2 of Touati & Ollivier (2021)). Analogously to USF, FB aligns the policy and reward spaces by conditioning a family of optimal policies on a task-specific latent z. However, FB departs from USF in its theoretical expressivity: whereas USF is constrained by a pre-defined set of basic features ϕ, FB, assuming sufficiently high d, can represent any reward function.

Training Training is performed by minimizing a loss similar to that of Eq. (16) where we additionally sample over the space of reward embeddings $\mathcal{Z}$.

$$L(F, B) = \mathbb{E}_{\substack{s, a \sim \rho, s^+ \sim \rho \\ z \sim p_{\mathcal{Z}}}}[(m_{F_z, B}(s, a, s^+) - \hat{m}_{\bar{F}_z, \bar{B}}(s, a, s^+))^2], \tag{20}$$

where $m_{F_z, B}$ is a parametric version of m^{π_z} and the target $\hat{m}_{\bar{F}_z, \bar{B}}(s, a, s^+) = \frac{\delta_s(\mathrm{d}s^+)}{\rho(\mathrm{d}s^+)} + \gamma \mathbb{E}_{s' \sim p(\cdot|s,a), a' \sim \pi_z(\cdot|s')}[\hat{m}_{\bar{F}_z, \bar{B}}(s', a', s^+)]$. In continuous spaces, the hard maximum over actions is replaced by either a softmax (Touati & Ollivier, 2021) or a z-conditioned policy network (Touati et al., 2023). To resolve the scale ambiguity in the factorization $F^\top B$ an auxiliary orthonormalization loss enforcing $\mathbb{E}_\rho[BB^\top] \approx I$ is added (Touati & Ollivier, 2021).

Relation to Framework In FB representations, the training target is the joint representation $\mu^r = (F, B)$. Due to the intrinsic coupling between policy and reward, training an FB representation constitutes a *pseudo reward-free* approach: although it aims to generalize across rewards, it still requires access to task embeddings during training in order to optimize its objective. For a given reward function r, its corresponding representation can be obtained as $z_r(B) = \mathbb{E}_{s\sim\rho}[r(s)B(s)]$, which induces the following decomposition:

$$\mathcal{F}(\mu = (F, B), r) = F(s, a, z_r(B))^\top z_r(B). \tag{21}$$

Policy inference is then performed accordingly via $\pi_r(\cdot|s) = \arg\max_a F(s, a, z_r(B))^\top z_r(B)$.

5.5 Proto Successor Measures

Proto Successor Measures (PSM; Agarwal et al. 2025b) take a reward-free approach to learning the space of successor measures. Extracting the optimal policy from such representation requires a search. PSM exploits properties of the SM to define a tractable search over such space.

The SM admits a linear decomposition $m^{\pi_y}(s, a, s^+) = \Phi(s, a, s^+)^\top y + b(s, a, s^+)$, where the basis function $\Phi : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^d$ and bias term $b : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ are policy-independent. The policy π_y is represented by the weight vector $y \in \mathbb{R}^d$. This linear decomposition allows to cast the zero-shot objective under the dual linear programming formulation (LP). Specifically, the objective is to find a y, that maximizes some reward function r. We can ensure the search is constrained to only valid successor measures by applying a non-negativity constraint, resulting in the following objective for extracting the optimal policy weights y^*:

$$y^* = \arg\max_y \mathbb{E}_{\substack{s\sim p_0, a\sim\pi_y(\cdot|s)\\ s^+\sim\rho}}[m^{\pi_y}(s, a, s^+)r(s^+)] \quad \text{s.t.} \quad m^{\pi_y}(s, a, s^+) \geq 0 \quad \forall (s, a, s^+), \tag{22}$$

The expectations over p_0 and π_y are intractable since we do not have direct access to these distributions, and are replaced by expectations over transitions from the dataset ρ, restricting coverage to the support of the behavior policy.

Training The components of the representation can be learned by minimizing an objective similar to Eq. (16). To allow sampling over policies, we define a discrete family of policies $(\pi_v)_{v\in\mathcal{V}}$, with $\mathcal{V} \subseteq \mathbb{Z}$; an additional parametric function $y : \mathbb{Z} \to \mathbb{R}^d$ maps each policy identifier onto a vector. The components are individually parametrized and learned by minimizing a reward free objective over the dataset of transitions and policies, resulting in the following loss

$$L(\Phi, y, b) = \mathbb{E}_{\substack{s,a\sim\rho,\, s^+\sim\rho\\ v\sim p_\mathcal{V}}}[(m_{\Phi,y(v),b}(s, a, s^+) - \hat{m}_{\bar{\Phi},\bar{y}(v),\bar{b}}(s, a, s^+))^2], \tag{23}$$

where $m_{\Phi,y(v),b}$ is a parametric version of $m^{\pi_{y(v)}}$ and the target $\hat{m}_{\bar{\Phi},\bar{y}(v),\bar{b}}(s, a, s^+) = \frac{\delta_s(\mathrm{d}s^+)}{\rho(\mathrm{d}s^+)} + \gamma\mathbb{E}_{s'\sim p(\cdot|s,a),a'\sim\pi_v(\cdot|s')}[\hat{m}_{\bar{\Phi},\bar{y}(v),\bar{b}}(s', a', s^+)]$.

Given a learned representation, the policy extraction phase requires solving the LP in Eq. (22), from which we obtain the SM of the optimal policy. For continuous action spaces, obtaining the policy requires learning a parameterized actor function.

Relation to Framework The representation in PSM does not exploit ties between rewards and policies, making this a reward-free, compositional method. This also allows the representation to be learned through Bellman evaluation backups, avoiding the possible instabilities caused by the maximization over the action-value function, typical of optimality backups. This methods exhibit the same decomposition operator as the SM, defined by Eq. (17). Based on this decomposition operator, policy extraction is performed via the search defined in Eq. (22), resulting in the optimal parameters y^* that define the extracted policy.

6 Related Work

The unified framework for unsupervised RL proposed by Agarwal et al. (2025a) is most closely related to our work. They establish a formal equivalence of several unsupervised RL objectives via the SM. Our framework, by contrast, focuses on structuring the landscape of zero-shot methods by analyzing the implications of various design choices, offering a complementary view.

Unsupervised RL methods (Jaderberg et al., 2017; Eysenbach et al., 2019; Laskin et al., 2021) undergo a pretraining phase (either in an reward-free or pseudo reward-free paradigm) followed by a finetuning phase to an extrinsic reward. Zero-shot RL is a subset of URL, where the finetuning phase is ignored and the agent is evaluated immediately after the pretraining phase.

Transfer learning in RL (TL), as surveyed in Taylor & Stone (2009), improves sample efficiency and generalization across tasks by reusing knowledge from prior experience. TL encompasses paradigms that adapt to changes in all parts of the MDP. Within this broader framework, zero-shot RL can be viewed as a form of transfer where adaptation is restricted to changes in the reward.

Model-based RL, as surveyed in Moerland et al. (2023), leverages transition dynamics models to plan by unrolling simulated trajectories (Janner et al., 2019). While decoupling dynamics from rewards enables flexibility, determining optimal values requires tracing potential futures. Model-based agents therefore adapt to new rewards through test-time search, while zero-shot RL demands fast generalization to novel objectives without explicit test-time planning.

7 Discussion & Conclusion

Our framework reveals an ambiguity in the definition of zero-shot RL: the absence of a standardized computational budget for policy extraction. While parameter updates to μ and explicit planning (i.e., reasoning over state transitions to synthesize new behaviors) are clearly prohibited, the allowable complexity of the operator $\mathcal{F}$ remains loosely specified. This is particularly evident in policy-conditioned, reward-free methods that search over policy space to maximize a given reward, where no clear boundary on search complexity exists. We argue that this reflects the inherent difficulty of defining a universal limit, effectively leaving the upper bound of what constitutes "zero-shot" to the practitioner. In practice, one may impose a fixed search budget and compare methods under this constraint. It is worth noting that pseudo reward-free methods admit a strict realization of zero-shot: parameterizing policies by the reward for which they are optimal, allows the optimal value function to be extracted by conditioning the representation on the desired reward.

More broadly, zero-shot methods share a common principle: behaviors are encoded in the learned representation and only extracted at test time, rather than synthesized through planning or adapted via finetuning.

This work also highlights several directions for future research. First, direct methods may benefit from advances in representation learning, where smooth and expressive reward embeddings could improve generalization (Bengio et al., 2013; Wu et al., 2018). In continuous spaces, greedy policy extraction may introduce out-of-distribution errors, suggesting regularization techniques as a potential remedy (Jeen et al., 2024; Zheng et al., 2025). Reward-free compositional methods instead rely on Bellman evaluation backups, offering a possible way to avoid such compounding errors. While most recent works focus on offline RL to isolate the representation learning problem, the online regime introduces exploration challenges; leveraging zero-shot representations for exploration remains a promising direction (Sun et al., 2025; Urpí et al., 2025). Finally, existing benchmarks such as URLB (Laskin et al., 2021) or ExoRL (Yarats et al., 2022) may obscure representation-specific limitations of zero-shot methods, motivating the development of dedicated benchmarks (Cetin et al., 2025).

References

Siddhant Agarwal, Caleb Chuck, Harshit Sikchi, Jiaheng Hu, Max Rudolph, Scott Niekum, Peter Stone, and Amy Zhang. A unified framework for unsupervised reinforcement learning algorithms. In *Workshop on Reinforcement Learning Beyond Rewards@ Reinforcement Learning Conference 2025*, 2025a.

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*, 2025b. URL https://openreview.net/forum?id=mUDnPzopZF.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

Marco Bagatella, Matteo Pirotta, Ahmed Touati, Alessandro Lazaric, and Andrea Tirinzoni. TD-JEPA: Latent-predictive representations for zero-shot reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=SzXDuBN8M1.

Andre Barreto, Will Dabney, Remi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor Features for Transfer in Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/hash/350db081a661525235354dd3e19b8c05-Abstract.html.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

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, Andre Barreto, John Quan, Daniel J. Mankowitz, Hado van Hasselt, Remi Munos, David Silver, and Tom Schaul. Universal successor features approximators. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=S1VWjiRcKX.

Edoardo Cetin, Ahmed Touati, and Yann Ollivier. Finer behavioral foundation models via auto-regressive features and advantage weighting. In *Reinforcement Learning Conference*, 2025. URL https://openreview.net/forum?id=LKtqc39P2v.

Yevgen Chebotar, Karol Hausman, Yao Lu, Ted Xiao, Dmitry Kalashnikov, Jake Varley, Alex Irpan, Benjamin Eysenbach, Ryan Julian, Chelsea Finn, et al. Actionable models: Unsupervised offline reinforcement learning of robotic skills. In *38th International Conference on Machine Learning, ICML 2021*, pp. 1518–1528. ML Research Press, 2021.

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

Benjamin Eysenbach, Julian Ibarz, Abhishek Gupta, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *7th International Conference on Learning Representations, ICLR 2019*, 2019.

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

Kevin Frans, Seohong Park, Pieter Abbeel, and Sergey Levine. Unsupervised zero-shot reinforcement learning via functional reward encodings. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=a6wCNfIj8E.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1587–1596. PMLR, 2018. URL https://proceedings.mlr.press/v80/fujimoto18a.html.

Steven Hansen, Will Dabney, Andre Barreto, David Warde-Farley, Tom Van de Wiele, and Volodymyr Mnih. Fast task inference with variational intrinsic successor features. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJeAHkrYDS.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. In *International Conference on Learning Representations*, 2017.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32, 2019.

Scott Jeen, Tom Bewley, and Jonathan Cullen. Zero-shot reinforcement learning from low quality data. *Advances in Neural Information Processing Systems*, 37:16894–16942, 2024.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=68n2s9ZJWF8.

Michael Laskin, Denis Yarats, Hao Liu, Kimin Lee, Albert Zhan, Kevin Lu, Catherine Cang, Lerrel Pinto, Pieter Abbeel, et al. Urlb: Unsupervised reinforcement learning benchmark. In *35th Conference on Neural Information Processing Systems (NeurIPS)*. Neural Information Processing Systems Foundation, 2021.

Andrew Levy, Robert Platt, and Kate Saenko. Hierarchical reinforcement learning with hindsight. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=ryzECoAcY7.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, 2016. preprint arXiv:1509.02971.

Hao Liu and Pieter Abbeel. Aps: Active pretraining with successor features. In *International Conference on Machine Learning*, pp. 6736–6747. PMLR, 2021.

Daniel J Mankowitz, Andrea Michi, Anton Zhernov, Marco Gelmi, Marco Selvi, Cosmin Paduraru, Edouard Leurent, Shariq Iqbal, Jean-Baptiste Lespiau, Alex Ahern, et al. Faster sorting algorithms discovered using deep reinforcement learning. *Nature*, 618(7964):257–263, 2023.

Azalia Mirhoseini, Anna Goldie, Mustafa Yazgan, Joe Wenjie Jiang, Ebrahim Songhori, Shen Wang, Young-Joon Lee, Eric Johnson, Omkar Pathak, Azade Nova, et al. A graph placement methodology for fast chip design. *Nature*, 594(7862):207–212, 2021.

Thomas M Moerland, Joost Broekens, Aske Plaat, Catholijn M Jonker, et al. Model-based reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 16(1):1–118, 2023.

Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. Hiql: Offline goal-conditioned rl with latent states as actions. *Advances in Neural Information Processing Systems*, 36:34866–34891, 2023.

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

Martin L Puterman. Markov decision processes: Discrete stochastic dynamic programming, 1994.

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

Jaemin Seo, SangKyeun Kim, Azarakhsh Jalalvand, Rory Conlin, Andrew Rothstein, Joseph Abbate, Keith Erickson, Josiah Wai, Ricardo Shousha, and Egemen Kolemen. Avoiding fusion plasma tearing instability with deep reinforcement learning. *Nature*, 626(8000):746–751, 2024.

RL Stratonovich. Conditional Markov processes. *Theory of Probability And Its Applications*, 5(2): 156–178, 1960.

Jingbo Sun, Songjun Tu, Haoran Li, Xin Liu, Yaran Chen, Ke Chen, Dongbin Zhao, et al. Unsupervised zero-shot reinforcement learning via dual-value forward-backward representation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Richard S. Sutton and Andrew Barto. *Reinforcement learning: an introduction*. Adaptive computation and machine learning. The MIT Press, Cambridge, Massachusetts London, England, second edition edition, 2020. ISBN 978-0-262-03924-6.

Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(7), 2009.

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.

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*, 2023. URL https://openreview.net/forum?id=MYEap_OcQI.

Núria Armengol Urpí, Marin Vlastelica, Georg Martius, and Stelian Coros. Epistemically-guided forward-backward exploration. In *Reinforcement Learning Conference*, 2025. URL https://openreview.net/forum?id=rJ0QbzBTRF.

Yifan Wu, George Tucker, and Ofir Nachum. The laplacian in RL: Learning representations with efficient approximations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJlNpoA5YQ.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3733–3742, 2018.

Sherry Yang, Ofir Nachum, Yilun Du, Jason Wei, Pieter Abbeel, and Dale Schuurmans. Foundation Models for Decision Making: Problems, Methods, and Opportunities, March 2023. URL http://arxiv.org/abs/2303.04129. arXiv:2303.04129 [cs].

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.

Chongyi Zheng, Royina Karegoudra Jayanth, and Benjamin Eysenbach. Can we really learn one representation to optimize all rewards? *arXiv preprint arXiv:2602.11399*, 2026.

Kexin Zheng, Lauriane Teyssier, Yinan Zheng, Yu Luo, and Xianyuan Zhan. Towards robust zero-shot reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=vbFZlM7nRK.

Supplementary Materials

The following content was not necessarily subject to peer review.

A Error Decompositions

In this section we analyze error bounds for various zero-shot methods under a common error decomposition. To enable a more principled comparison, we extend existing error bounds to better illustrate how structural assumptions induce distinct but related error terms.

All previously introduced methods are theoretically well-founded and recover the optimal value function in the idealized setting via $Q_r^* = \mathcal{F}(\mu, r)$. While optimality is theoretically attainable, practical implementations – constrained by factors such as finite model capacity, restricted search spaces, and approximate expectations – incur three primary sources of error:

- *Inference Error:* In some cases it might not be possible to exactly evaluate the decomposition operator. For instance, when $\mathcal{F}$ requires a search over the full policy space, such as in SF&GPI.
- *Reward Error:* Even with access to the true reward function, reward embedding-based methods may suffer from errors introduced by the latent representation itself.
- *Approximation Error:* We always recover an approximate variant of the true μ, due to limited data and computational resources.

Informally, given our extracted policy $\tilde{\pi} = \arg\max_a \tilde{\mathcal{F}}(\tilde{\mu}, \tilde{r})$ we establish error bounds of the generic form

$$\|Q_r^* - Q_r^{\tilde{\pi}}\| \leq C_1 \underbrace{\|\tilde{\mathcal{F}} - \mathcal{F}\|}_{\varepsilon_{\text{inference}}} + C_2 \underbrace{\|\tilde{r} - r\|}_{\varepsilon_{\text{reward}}} + C_3 \underbrace{\|\tilde{\mu} - \mu\|}_{\varepsilon_{\text{approx}}}. \tag{24}$$

Detailed proofs, constants, and norm definitions are provided in Appendix C.

Sucessor Features & GPI We extend the results of (Barreto et al., 2017; Borsa et al., 2019) to account for approximate reward linearizations:

Theorem A.1. *Let ϕ be a basic feature mapping, let $\{\pi_i\}_{i=1}^n$ be a set of policies, and let $\{w_i\}_{i=1}^n$ be the corresponding linear weights such that π_i is optimal on $\phi^\top w_i$. Let $\tilde{\psi}_i(s,a)$ denote the learned SF for the given policies. Let r be an arbitrary reward function and w_r its approximate linearization such that $r \approx \phi^\top w_r$, and let $\pi(\cdot|s) = \arg\max_a \max_{i\in[n]} \tilde{\psi}_i(s,a)^\top w_r$ be the GPI policy. Then there exists constants $C_{\gamma,\phi}, C_\gamma, C_r \geq 0$ such that*

$$\|Q_r^* - Q_r^\pi\|_\infty \leq C_{\gamma,\phi} \min_{i\in[n]} \|w_r - w_i\| + C_\gamma \|r - \phi^\top w_r\| + C_r \max_{i\in[n]} \|\psi^{\pi_i}(\cdot) - \tilde{\psi}_i(s,a)\|. \tag{25}$$

The first term reflects a restricted policy search: unless an optimal policy for the target reward is contained in the training set, an error during policy extraction is unavoidable. This error can be expected to decrease as the number of trained policies grows. The second term captures reward linearization error, which vanishes only if rewards are exactly linear in ϕ. The final term corresponds to approximation error in learning successor features.

Universal Successor Features Applying the same reasoning to USFs yields:

Theorem A.2. *Let ϕ be a basic feature mapping, let $\tilde{\psi}(\cdot, w)$ denote the learned USFs for some $w \in \mathcal{W}$. Let r be an arbitrary reward function and $w_r \in \mathcal{W}$ its approximate linearization such that $r \approx \phi^\top w_r$, and let $\pi(\cdot|s) = \arg\max_a \tilde{\psi}(s,a,w_r)^\top w_r$ be the extracted policy. Let $\psi^{w_r}(\cdot)$ denote the SFs for the policy optimal on $\phi^\top w_r$. Then:*

$$\|Q_r^* - Q_r^\pi\|_\infty \leq C_\gamma \|r - \phi^\top w_r\| + C_r \|\psi^{w_r}(\cdot) - \tilde{\psi}(\cdot, w_r)\|. \tag{26}$$

As in SF&GPI, approximate linearization induces reward error. In contrast to SF&GPI, USFs remove explicit inference error by parameterizing policies through reward weights. However, generalization across $\mathcal{W}$ may increase approximation error, especially in regions where coverage is limited.

Forward-Backward Representations We extend results from (Touati & Ollivier, 2021) by incorporating approximation error:

Theorem A.3. *Let F, B a forward-backward representation, such that Eq. (19) holds for some $\rho \in \Delta(\mathcal{S})$ and let $\tilde{F}, \tilde{B}$ denote their learned versions. Let r be some reward function, let $z_r = \mathbb{E}_{s\sim\rho}[B(s)r(s)]$ and let $\pi(\cdot|s) = \arg\max_a \tilde{F}(s,a,z_r)^\top z_r$, then*

$$\|Q_r^* - Q_r^\pi\|_\infty \leq C_{r,\gamma}\left(\|\varepsilon_{\pi_{z_r},z_r}\| + \|\varepsilon_{\pi,z_r}\|\right) + \tilde{C}_{r,\gamma}\left(\|F(\cdot,z_r) - \tilde{F}(\cdot,z_r)\| + \|F^\top B - \tilde{F}^\top \tilde{B}\|\right). \tag{27}$$

where $\varepsilon_{\pi,z}(s,a,s^+) = m^\pi(s,a,s^+) - F(s,a,z)^\top B(s^+)$.

Unlike USF, FB does not impose a linear decomposition of the reward function a priori, and therefore introduces no explicit reward approximation error of the form in Eqs. 25 and 26. Instead, FB incurs an inference error arising from the structural assumption that the SM admits a factorization into F and B. Although this inference error is conceptually distinct from the linearization error in USFs, the two are closely related (see Touati & Ollivier 2021 for a detailed discussion). FB inevitably incurs approximation errors due to finite data and limited model capacity.

PSM likely admits a similarly-composed error bound to FB, where the inference would also be tied to the soft LP.

Direct Methods Direct approaches do not rely on value-function decomposition and therefore incur no decomposition-induced inference error:

Theorem A.4. *Let $f : \mathcal{R} \to \mathcal{Z} \subset \mathbb{R}^d$ be reward embedding. Let $\tilde{Q}_z$ denote some trained direct representation. Let r be a reward function, $f(r)$ its reward embedding, and let $r_f \in \mathcal{R}$ be a reward function such that $Q^*_{r_f}$ is the true optimal value function corresponding to the embedding $f(r)$. Then the value error of direct methods decomposes into*

$$\|Q_r^* - \tilde{Q}_{f(r)}\|_\infty \leq \frac{1}{1-\gamma}\|r - r_f\| + \|Q^*_{r_f} - \tilde{Q}_{f(r)}\|. \tag{28}$$

The error of direct methods decomposes into two components. The first captures the effect of imperfect reward embeddings: even if optimization were exact, the reward function r_f induced by the embedding may differ from the true reward r, leading to a systematic mismatch. The second component corresponds to the approximation error.

B Feature Matrix of Zero-Shot Methods

Table 1: Feature matrix of a *non-exhaustive* list of zero-shot RL methods

Method	Representation	Reward	Paradigm	Reward Embedding	Policy Extraction	Linear Constraint on Reward
SF+GPI (Barreto et al., 2017)	Compositional	Free	Online	Lin. Reg.	GPI	Yes
USF (Borsa et al., 2019)	Compositional	Pseudo	Online	Lin. Reg.	Forward pass	Yes
USF Lap. (Wu et al., 2019)	Compositional	Pseudo	Online	Lin. Reg.	Forward pass	Yes
FB (Touati & Ollivier, 2021; Touati et al., 2023)	Compositional	Pseudo	Offline	MC Estimate	Forward pass	Yes
FB-AWARE (Cetin et al., 2025)	Compositional	Pseudo	Offline	Autoregressive	Forward pass	No
VISR (Hansen et al., 2020)	Compositional	Pseudo	Online	Lin. Reg.	Forward pass	Yes
APS (Liu & Abbeel, 2021)	Compositional	Pseudo	Online	Lin. Reg.	Forward pass	Yes
PSM (Agarwal et al., 2025b)	Compositional	Free	Offline	N/A	Constrained LP	No
FRE (Frans et al., 2024)	Direct	Pseudo	Offline	Forward pass	Forward pass	No
HILP (Park et al., 2024)	Direct	Pseudo	Offline	Lin. Reg.	Forward pass	Yes
FBEE^Q (Urpí et al., 2025)	Compositional	Pseudo	Online	MC Estimate	Forward pass	Yes
DVFB (Sun et al., 2025)	Compositional	Pseudo	Online	MC Estimate	Forward pass	Yes
VC-FB (Jeen et al., 2024)	Compositional	Pseudo	Online	MC Estimate	Forward pass	Yes
BREEZE (Zheng et al., 2025)	Compositional	Pseudo	Offline	MC Estimate	Forward pass	Yes
one-step FB (Zheng et al., 2026)	Compositional	Free	Offline	MC Estimate	Forward pass	Yes
TD-JEPA (Bagatella et al., 2026)	Compositional	Free	Offline	Lin. Reg.	Forward pass	Yes

C Proofs

In the following we will provide proofs for the statements in section A.

To start with, we want to clarify, that for convenience of notation we will abuse notation and write $\sup_{(s,a)\in\mathcal{S}\times\mathcal{A}} |f(s,a)| = \|f\|_\infty$, even though, depending on f and $\mathcal{S}\times\mathcal{A}$ this may not be a well-defined norm.

C.1 Theorem A.1

Theorem C.1 (Theorem A.1)**.** *Let ϕ be a basic feature mapping, let $\{\pi_i\}_{i=1}^n$ be a set of policies, and let $\{w_i\}_{i=1}^n$ be the corresponding linear weights such that π_i is optimal on $\phi^\top w_i$. Let $\tilde{\psi}_i(s,a)$ denote the learned SF for the given policies. Let r be an arbitrary reward function and w_r its approximate linearization such that $r \approx \phi^\top w_r$, and let $\pi(\cdot|s) = \arg\max_a \max_{i\in[n]} \tilde{\psi}_i(s,a)^\top w_r$ be the GPI policy. Then*

$$\|Q_r^* - Q_r^\pi\|_\infty \leq \frac{2}{1-\gamma} \min_{i\in[n]} \|\phi\|_\infty \|w_r - w_i\| + \frac{2}{1-\gamma}\|r - \phi^\top w_r\|_\infty + \max_{i\in[n]} \|w_r\| \|\psi^{\pi_i}(\cdot) - \tilde{\psi}_i(s,a)\|_\infty$$

Proof. We can decompose the optimality gap as

$$\begin{aligned}\|Q_r^* - Q_r^\pi\|_\infty &\leq \|Q_r^* - Q^*_{\phi^\top w_r}\|_\infty + \|Q^*_{\phi^\top w_r} - Q^\pi_{\phi^\top w_r}\|_\infty + \|Q^\pi_{\phi^\top w_r} - Q_r^\pi\|_\infty \\ &\leq 2\max_{\tilde{\pi}} \|Q^{\tilde{\pi}}_{\phi^\top w_r} - Q_r^{\tilde{\pi}}\|_\infty + \|Q^*_{\phi^\top w_r} - Q^\pi_{\phi^\top w_r}\|_\infty. \end{aligned} \tag{29}$$

Here the first inequality follows from the triangle inequality. Lastly, by Theorem 1 in Borsa et al. (2019) we have

$$\|Q^*_{\phi^\top w_r} - Q^\pi_{\phi^\top w_r}\|_\infty \frac{2}{1-\gamma} \min_{i\in[n]} \|\phi\|_\infty \|w_r - w_i\| + \max_{i\in[n]} \|w_r\| \|\psi^{\pi_i}(\cdot) - \tilde{\psi}_i(s,a)\|_\infty \tag{30}$$

For any policy π and two reward functions r, r' we can estimate the gap of the value functions using the result of Proposition 17 from (Touati & Ollivier, 2021):

$$\|Q_r^\pi - Q_{r'}^\pi\|_\infty \leq \frac{1}{1-\gamma}\|r - r'\|_\infty. \tag{31}$$

This, in combination with Eq. (30) and (29) completes the proof.

□

Note, that in finite MDPs, we can also use the euclidean norm to bound this value-gap, instead of using the supremum: For any policy π, and reward functions r, r' we have

$$\|Q_r^\pi - Q_{r'}^\pi\|_\infty \leq \frac{\|r - r'\|_2}{1-\gamma}. \tag{32}$$

Here r, r' are treated as vectors in $\mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$, and $\|\cdot\|_2$ denotes the euclidean norm. Notably, for $r' = \phi^\top w_r$ this distance is indeed minimized by obtaining w_r that minimizes the mean squared error, which can be obtained by sampling uniformly in Eq. (10).

Proof. For convenience of notation we write $\Delta r(\cdot) = r(\cdot) - r'(\cdot)$. Moreover, let $\rho^\pi(\cdot|t,s,a) \in \Delta(\mathcal{S}\times\mathcal{A})$ denote the state-action distribution under π at timestep t starting in $s_0 = s, a_0 = a$. As we are in the finite MDP setting we will treat Δr and $\rho^\pi(\cdot|t,s,a)$ as objects in $\mathbb{R}^{|\mathcal{S}|\times|\mathcal{A}|}$, and denote by $\langle\cdot,\cdot\rangle$ the standard inner product on this space.

For some policy π and an arbitrary state action pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ we have

$$\begin{aligned}
&|Q_r^\pi(s,a) - Q_{r'}^\pi(s,a)| \\
&\leq \sum_{t\geq 0} \gamma^t |\langle \rho^\pi(\cdot|t,s,a), \Delta r(\cdot)\rangle| \\
&\leq \sum_{t\geq 0} \gamma^t \|\rho^\pi(\cdot|t,s,a)\|_2 \|\Delta r\|_2 \\
&\leq \|\Delta r\|_2 \sum_{t\geq 0} \gamma^t \|\rho^\pi(\cdot|t,s,a)\|_2 \\
&\leq \|\Delta r\|_2 \frac{1}{1-\gamma}.
\end{aligned}$$

Here the second inequality follows from Cauchy-Schwarz inequality, and the final one from Cauchy-Schwarz and the fact that the distribution has bounded mass:

$$\|\rho^\pi(\cdot|t,s,a)\|_2 \leq \|1\|_\infty \|\rho^\pi(\cdot|t,s,a)\|_1 \leq 1.$$

The established expression is independent of the chosen state-action pair completing the proof. □

Lastly, it should be noted, that Theorem A.2 is a direct consequence of the result above.

C.2 Theorem A.3

Theorem C.2 (Theorem A.3)**.** *Let F, B a forward-backward representation, such that Eq. (19) holds for some $\rho \in \Delta(\mathcal{S})$ and let $\tilde{F}, \tilde{B}$ denote their learned versions. Let r be some reward function,let $z_r = \mathbb{E}_{s\sim\rho}[B(s)r(s)]$ and let $\pi(\cdot|s) = \arg\max_a \tilde{F}(s,a,z_r)^\top z_r$, then*

$$\begin{aligned}
&\|Q_r^* - Q_r^\pi\|_\infty \\
&\leq C_1 \left(\sup_{s,a} \|\varepsilon_{\pi_{z_r},z_r}(s,a,\cdot)\|_{L^1(\rho)} + \sup_{s,a} \|\varepsilon_{\pi,z_r}(s,a,\cdot)\|_{L^1(\rho)} \right) \\
&+ C_2 \sup_{s,a} \|F(s,a,z_r) - \tilde{F}(s,a,z_r)\|_2 + \sup_{s,a} \|F(s,a,z_r)^\top B(\cdot) - \tilde{F}(s,a,z_r)^\top \tilde{B}(\cdot)\|_{L^1(\rho)}
\end{aligned} \tag{33}$$

where $\varepsilon_{\pi,z}(s,a,s^+) = m^\pi(s,a,s^+) - F(s,a,z)^\top B(s^+)$.

Proof. Let everything be defined as in the assumption for the theorem. We can firstly decompose the error as

$$\|Q_r^* - Q_r^\pi\|_\infty \leq \|Q_r^* - Q_r^{\pi_{z_r}}\|_\infty + \|Q_r^{\pi_{z_r}} - Q_r^\pi\|_\infty. \tag{34}$$

Following Touati & Ollivier (2021), we define $\varepsilon_{\pi,z}(s,a,s^+) = m^\pi(s,a,s^+) - F(s,a,z)^\top B(s^+)$ and $\tilde{\varepsilon}_{\pi,z}(s,a,s^+) = m^\pi(s,a,s^+) - \tilde{F}(s,a,z)^\top \tilde{B}(s^+)$. Using this, the first term can be bounded as

$$\|Q_r^* - Q_r^{\pi_{z_r}}\|_\infty \leq \frac{3\|r\|_\infty}{1-\gamma} \sup_{s,a} \|\varepsilon_{\pi_{z_r},z_r}(s,a,\cdot)\|_{L^1(\rho)}. \tag{35}$$

Here $\|\cdot\|_{L^1(\rho)}$ denotes the L1-Norm. The second term can now be further decomposed as

$$\|Q_r^{\pi_{z_r}} - Q_r^\pi\|_\infty \leq \|Q_r^{\pi_{z_r}} - F(\cdot,z_r)^\top z_r\|_\infty + \|Q_r^\pi - \tilde{F}(\cdot,z_r)^\top z_r\|_\infty + \|F(\cdot,z_r)^\top z_r - \tilde{F}(\cdot,z_r)^\top z_r\|_\infty.$$

The first component on the right hand side can again be bounded by results from (Touati & Ollivier, 2021):

$$\|Q_r^{\pi_{z_r}} - F(\cdot,z_r)^\top z_r\|_\infty \leq \|r\|_\infty \sup_{s,a} \|\varepsilon_{\pi_{z_r},z_r}(s,a,\cdot)\|_{L^1(\rho)}. \tag{36}$$

The same result can be applied to bound the second term as

$$\|Q_r^\pi - \tilde{F}(\cdot, z_r)^\top z_r\|_\infty \leq \|r\|_\infty \sup_{s,a} \|\tilde{\varepsilon}_{\pi,z_r}(s,a,\cdot)\|_{L^1(\rho)}. \tag{37}$$

Lastly, the last component can be bounded by Cauchy-Schwartz as

$$\|F(\cdot, z_r)^\top z_r - \tilde{F}(\cdot, z_r)^\top z_r\|_\infty = \sup_{s,a} |F(s,a,z_r)^\top z_r - \tilde{F}(s,a,z_r)^\top z_r| \tag{38}$$

$$\leq \|z_r\|_2 \sup_{s,a} \|F(s,a,z_r) - \tilde{F}(s,a,z_r)\|_2, \tag{39}$$

where $\|\cdot\|_2$ denotes the euclidean norm. Gluing it together yields

$$\|Q_r^* - Q_r^\pi\|_\infty \tag{40}$$

$$\leq C_{r,\gamma} \left(\sup_{s,a} \|\varepsilon_{\pi_{z_r},z_r}(s,a,\cdot)\|_{L^1(\rho)} + \sup_{s,a} \|\tilde{\varepsilon}_{\pi,z_r}(s,a,\cdot)\|_{L^1(\rho)} \right) \tag{41}$$

$$+ C_r \sup_{s,a} \|F(s,a,z_r) - \tilde{F}(s,a,z_r)\|_2. \tag{42}$$

Where we defined $C_{r,\gamma} := \left(\frac{3}{1-\gamma} + 1\right) \|r\|_\infty$, and $C_r := \|z_r\|$. Lastly, we decompose $\|\varepsilon_\pi(s,a,\cdot)\|_{L^1(\rho)}$ into terms that we can attribute to the inference and approximation error respectively. By the triangle inequality we have

$$\|\varepsilon_{\pi,z_r}(s,a,\cdot)\|_{L^1(\rho)} \tag{43}$$

$$\leq \|m^\pi(s,a,\cdot) - F(s,a,z_r)^\top B(\cdot)\|_{L^1(\rho)} + \|F(s,a,z_r)^\top B(\cdot) - \tilde{F}(s,a,z_r)^\top \tilde{B}(\cdot)\|_{L^1(\rho)} \tag{44}$$

$$= \|\varepsilon_{\pi,z_r}(s,a,\cdot)\|_{L^1(\rho)} + \|F(s,a,z_r)^\top B(\cdot) - \tilde{F}(s,a,z_r)^\top \tilde{B}(\cdot)\|_{L^1(\rho)}. \tag{45}$$

Combining this with Eqs. 40 - 42 yields some constants C_1, C_2 only dependent on r and γ such that

$$\|Q_r^* - Q_r^\pi\|_\infty \tag{46}$$

$$\leq C_1 \left(\sup_{s,a} \|\varepsilon_{\pi_{z_r},z_r}(s,a,\cdot)\|_{L^1(\rho)} + \sup_{s,a} \|\varepsilon_{\pi,z_r}(s,a,\cdot)\|_{L^1(\rho)} \right) \tag{47}$$

$$+ C_2 \sup_{s,a} \|F(s,a,z_r) - \tilde{F}(s,a,z_r)\|_2 + \sup_{s,a} \|F(s,a,z_r)^\top B(\cdot) - \tilde{F}(s,a,z_r)^\top \tilde{B}(\cdot)\|_{L^1(\rho)} \tag{48}$$

□

C.3 Theorem A.4

Theorem C.3 (Theorem A.4)**.** *Let $f : \mathcal{R} \to \mathcal{Z} \subset \mathbb{R}^d$ be reward embedding. Let $\tilde{Q}_z$ denote some trained direct representation. Let r be a reward function, $f(r)$ its reward embedding, and let $r_f \in \mathcal{R}$ be a reward function such that $Q^*_{r_f}$ is the true optimal value function corresponding to the embedding $f(r)$. Then the value error of direct methods decomposes into*

$$\|Q_r^* - \tilde{Q}_{f(r)}\|_\infty \leq \frac{1}{1-\gamma} \|r - r_f\|_\infty + \|Q^*_{r_f} - \tilde{Q}_{f(r)}\|_\infty \tag{49}$$

Proof. The proof is immediate as we can decompose

$$\|Q_r^* - \tilde{Q}_{f(r)}\|_\infty \leq \|Q_r^* - Q^*_{r_f}\|_\infty + \|Q^*_{r_f} - \tilde{Q}_{f(r)}\|_\infty. \tag{50}$$

The result then follows immediately from Eq. (31). □

D Additional Details Successor Measure

The successor measure M^π satisfies a Bellman-like equation (Blier et al., 2021):

$$M^\pi(s,a,X) = \mathbb{1}\{(s,a)\in X\} + \gamma\, \mathbb{E}_{s'\sim p(\cdot|s,a),\, a'\sim\pi(\cdot|s')}\left[M^\pi(s',a',X)\right]. \tag{51}$$

D.1 Parametrized SM loss

Norm on M:

$$||M||_\rho^2 = \mathbb{E}_{\substack{(s,a)\sim\rho\\ s^+\sim\rho}}\Big[\Big(\frac{M(s,a,\mathrm{d}s^+)}{\rho(\mathrm{d}s^+)}\Big)^2\Big] \tag{52}$$

parametric model of M:

$$m_\theta(s,a,\mathrm{d}s^+) = m_\theta(s,a,s^+)\rho(\mathrm{d}s^+) \tag{53}$$

The SM loss is derived by expanding the following TD error:

$$||M - M^{\text{target}}||_\rho^2 = ||M - (I+\gamma P\bar{M})||_\rho^2 \tag{54}$$

$$= ||m_\theta(s,a,s^+)\rho(\mathrm{d}s^+) - \delta_{(s)}(\mathrm{d}s^+) - \mathbb{E}_{\substack{s'\sim P(\mathrm{d}s'|s,a)\\ a'\sim\pi}}\left[\bar{m}_\theta(s',a',s^+)\rho(\mathrm{d}s^+)\right]||_\rho^2 \tag{55}$$

$$= \mathbb{E}_{\substack{(s,a)\sim\rho\\ s^+\sim\rho}}\big[\big(m_\theta(s,a,s^+) - \frac{\delta_{(s)}(\mathrm{d}s^+)}{\rho(\mathrm{d}s^+)} - \mathbb{E}_{\substack{s'\sim P(\mathrm{d}s'|s,a)\\ a'\sim\pi}}\left[\bar{m}_\theta(s',a',s^+)\right]\big)^2\big] \tag{56}$$

$$= \mathbb{E}_{\substack{(s,a)\sim\rho\\ s^+\sim\rho}}\big[\big(\overbrace{m_\theta(s,a,s^+)}^{b} \underbrace{- \frac{\delta_{(s)}(\mathrm{d}s^+)}{\rho(\mathrm{d}s^+)}}_{a} \overbrace{- \mathbb{E}_{\substack{s'\sim P(\mathrm{d}s'|s,a)\\ a'\sim\pi}}\left[\bar{m}_\theta(s',a',s^+)\right]}^{b}\big)^2\big] \tag{57}$$

Expand $(a+b)^2$ and move terms independent of m_θ into Const

$$= -2\mathbb{E}_{\substack{(s,a)\sim\rho\\ s^+\sim\rho}}\big[\underbrace{\frac{\delta_{(s)}(\mathrm{d}s^+)}{\rho(\mathrm{d}s^+)} m_\theta(s,a,s^+)}_{\text{Nonzero only if } s\in\mathrm{d}s^+}\big] + \mathbb{E}_{\substack{(s,a,s')\sim\rho, a'\sim\pi(\cdot|s')\\ s^+\sim\rho}}\left[(m_\theta(s,a,s^+) - \bar{m}_\theta(s',a',s^+))^2\right] + \text{Const} \tag{58}$$

$$= -2\mathbb{E}_{(s,a)\sim\rho}[m_\theta(s,a,s)] + \mathbb{E}_{\substack{(s,a,s')\sim\rho, a'\sim\pi(\cdot|s')\\ s^+\sim\rho}}\left[(m_\theta(s,a,s^+) - \bar{m}_\theta(s',a',s^+))^2\right] + \text{Const} \tag{59}$$

D.2 FB Loss with Successor Representation 1-shift

In Touati et al. (2023), the successor measure, represented as $M^\pi = F^\top B\rho$, is learned by minimizing a slightly different loss to the one in Touati & Ollivier (2021) which is the one we present in Section 5.4. This is due to a different definition of the successor measure itself in Touati et al. (2023), specifically they ignore the immediate occupancy term by multiplying the SM by the transition matrix P. In other words we check if the next state and action are in X rather than the current state and action. Specifically:

$$\tilde{M}^\pi(s,a,X) = \mathbb{E}_{\pi,p}\big[\sum_{t=0}^{\infty}\gamma^t \mathbb{1}\{(s_{t+1},a_{t+1})\in X\} \mid s_0 = s, a_0 = a\big]. \tag{60}$$

Or in matrix form

$$\tilde{M}^\pi = PI + \gamma P_\pi \tilde{M} \tag{61}$$

In words, the successor measure defined as in the above equations ignores the immediate occupancy (i.e. $\mathbb{1}\{(s_0,a_0)\in X\}$). The relationship between the successor measure M, as defined in Eq. (15), and $\tilde{M}$ is:

$$\tilde{M}^\pi = P^\pi(I + \gamma P^\pi M) \tag{62}$$

Resulting in the following loss for $\tilde{M} = F^\top B\rho$

$$\begin{aligned}\mathcal{L}(F,B) = &\mathbb{E}_{\substack{(s_t,a_t,s_{t+1})\sim\rho \\ (s',a')\sim\rho}}[(F(s_t,a_t,z)^\top B(s'a') - \bar{F}(s_{t+1},\pi_z(s_{t+1}),z)^\top \bar{B}(s',a'))^2] \\ &- 2\mathbb{E}_{(s_t,a_t,s_{t+1},a_{t+1})\sim\rho}[F(s_t,a_t,z)^\top B(s_{t+1},a_{t+1})].\end{aligned} \tag{63}$$

E Notation

We generally assume that any reward function $r \in \mathcal{R}$ is a mapping $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}, (s,a,s') \mapsto r(s,a,s')$. In some MDPs, the dependence on a or s' entirely drops, as for instance in goal-conditioned RL. When full dependence is present, we use the following short-hand notations by marginalizing over action and future state distributions

$$\begin{aligned} r(s,a) &= \mathbb{E}_{s'\sim p(\cdot|s,a)}[r(s,a,s')], \\ r(s) &= \mathbb{E}_{a\sim\pi(\cdot|s),s'\sim p(\cdot|s,a)}[r(s,a,s')]. \end{aligned}$$