Regularized Latent Dynamics Prediction is a Strong Baseline For Behavioral Foundation Models

Pranaya Jajoo[1,2], **Harshit Sikchi**[4,*], **Siddhant Agarwal**[4,*], **Amy Zhang**[4], **Scott Niekum**[5], **Martha White**[1,2,3]
pranayajajoo@ualberta.ca

[1]**Department of Computing Science, University of Alberta, Canada**
[2]**Alberta Machine Intelligence Institute (Amii)**
[3]**CIFAR AI Chair**
[4]**The University of Texas at Austin**
[5]**UMass Amherst**

Abstract

Behavioral Foundation Models (BFMs) have seen some success recently in producing agents with the capabilities to adapt to any unknown reward or task. In reality, these methods are only able to produce near-optimal policies for the reward functions that are in the span of some pre-existing *state features*. Naturally, their efficiency relies heavily on the choice of state features used by them. As a result, these BFMs have used a wide variety of potentially complex objectives to train task spanning features with different inductive properties. With this work, our aim is to examine the question: are these complex representation learning objectives necessary for zero-shot RL? Specifically, we revisit the objective of self-supervised next-state prediction for state feature learning, but observe that such an objective is prone to increasing state-feature similarity, and subsequently reducing span of reward functions that we can represent optimal policies for. We show that by simply maintaining feature diversity using orthonormal regularization along with next-state prediction, we can match or surpass state-of-the-art complex representation learning methods for zero-shot RL.

1 Introduction

The reward hypothesis states that all goals and purposes can be understood as maximization of scalar reward signals. This principle has motivated development of RL algorithms that learn efficiently given a reward function. However, a large part of prior developments in RL focus on dealing with single reward function or a small subset of reward functions. But with the recent focus on generalist agents, the generalization capabilities of RL to new tasks are being tested. Still, when compared to the supervised ML counterparts, RL lags behind in showing zero-shot generalization to new tasks in an environment.

Zero shot learning has been adapted in RL (Touati et al., 2023) to learn agents that can solve *any* task in the environment without any additional training or planning, after an initial pretraining. Zero-shot RL has significant practical potential in developing generalist agents with wide applicability. For instance, robotics applications, like robotic manipulation or drone navigation, often require agents to solve a wide variety of unknown tasks. A general-purpose household robot needs to possess the capability to flexibly adapt to various household chores without explicit training for each new task.

Behavioral foundation models based on the idea of leveraging successor representations (Touati et al., 2023; Agarwal et al., 2024) have shown promising progress towards developing algorithms

that output near-optimal policies for a wide class of reward functions without additional learning or training during test-time by pretraining on a dataset of reward-free interactions, i.e, zero-shot RL. Such BFMs work by a) learning a state representation $\phi : s \to \mathbb{R}^d$ and b) learning a space of policies parameterized by a latent vector $z \in \mathbb{R}^d$ trained to be optimal for reward defined as $r(s) = \phi(s)^\top.z$. At test time given any reward function $r^{test}(s)$, the near-optimal policy $\pi_{z_{r^{test}}}$ is obtained by projecting reward functions into the space of state-representations, and solving for $z_{r^{test}}$ such that $r^{test}(s) \approx \phi(s)^\top.z_{r^{test}}$. The near-optimal policy is then given by $\pi_{z_{r^{test}}}$.

The success of modern zero-shot RL methods is often attributed to learning generalizable state-representations. State-of-the-art methods usually learn state representations that retain information suitable to represent successor measures under a wide class of policies. Successor measures are information rich objects that capture a policy's state visitation in the environment given any starting state. Successor measures are usually learned for an explicitly defined class of policies (Agarwal et al., 2024) or implicitly by first defining a class of reward functions (Ramesh et al., 2021; Park et al., 2024) and considering optimal policies for those reward functions as the set of policies. The main insight behind predicting successor measure as a target for state representation learning is that representations sufficient to explain future state-visitation for a wide range of policies captures features that are relevant for sequential decision making under various reward functions.

Unfortunately, state representation learning by estimating successor measures requires iteratively applying Bellman evaluation backups or Bellman optimality backups, both of which are known to result in optimization difficulties or feature collapse due to the instability inherent in bootstrapping (Kumar et al., 2021). On the other hand, the dynamics learning objective is an optimization-friendly objective for state representation learning that bypasses bootstrapping. However, using the learned dynamics model to obtain a policy at test time would require a policy training phase with model based RL algorithm going against our objective of zero-shot RL. This work aims to investigate the following question:

Is next-state prediction enough to learn state features that enable performant zero-shot RL?

Our investigation is inspired by the work of Fujimoto et al. (2025), which underscored the importance of auxiliary objective of state representation learning through dynamics prediction losses in boosting performance of single-task model-free RL. Our work differs by tackling a different setting – we present an empirical investigation of the simple dynamics prediction objective for learning representations suitable for zero-shot RL. Unlike the single task RL setting examined by Fujimoto et al. (2025), we find that in its naive form, this objective leads to a mild form of feature collapse where representation of different states increase in similarity over training steps and result in poor zero-shot RL performance when evaluated on a number of downstream tasks. With a simple regularization to prevent collapse, we show that model-based representations learned via supervised learning are competitive and present a scalable alternative to representations learned via complex successor measure estimation methods for zero-shot RL.

2 Related Work

Unsupervised RL: Unsupervised RL encompasses the class of algorithms that enable learning general-purpose skills and representations without relying on reward signal in the data. In this work, we focus on techniques that learn representations capable of producing optimal value functions for any arbitrary function reward specification.

Recent pre-training approaches (e.g., Ma et al. (2023); Nair et al.) borrow self-supervised techniques from computer vision—such as masked auto-encoding—to extract embeddings from large-scale datasets (Grauman et al. (2022)) that can be fine-tuned for downstream control. However, these representations are inherently tied to the behavior policies used during data collection. These policies are limited in their ability to capture the full spectrum of possible behaviors or to approximate Q-functions for any reward functions. HILP (Park et al. (2024)) goes beyond standard masked autoencoding approaches by using Hilbert-space representations to preserve temporal dynamics.

Auxiliary objectives, which involve complementary predictive tasks to get richer semantic or temporal structures, have also been explored in previous works (Agarwal et al. (2021), Schwarzer et al. (2020)). Although representations from auxiliary objectives can accelerate policy learning, a new policy still needs to be learned from scratch for each new reward function.

Several works have also focused on intent or skill discovery through diversity-driven objectives. These methods consider state-visitation distribution that are defined by latents or skills. Thus, maximizing mutual information (Warde-Farley et al. (2018), Eysenbach et al. (2018), Achiam et al. (2018), Eysenbach et al. (2022)) or minimizing the Wasserstein distance (Park et al. (2023)) between latents and state-visitation distribution is used to ensure diversity.

Behavioral Foundation Models: Behavioral Foundation Models deals with the class of approaches that can be used to train an RL agent in an unsupervised manner using task-agnostic reward-free offline transitions. During inference, BFMs can approximate the optimal policy for a wide class of unseen reward functions without any further training.

Forward-Backward representations (Touati & Ollivier (2021)) and PSM (Agarwal et al. (2024)) provide a robust framework for BFMs based on stationary distribution, on which several successive works are based. Fast Imitation with BFMs (Pirotta et al. (2023)) demonstrates the ability of successor-measure–based BFMs to imitate new behaviors from just a few demonstrations, while Fast Adaptation with BFMs (Sikchi et al. (2025)) builds upon this by fine-tuning BFMs' latent embedding space, yielding 10-40% improvement over their zero-shot performance in a few of episodes. Recent progress in imitation learning has led to the development of BFMs tailored for humanoid control tasks (Peng et al. (2022), Won et al. (2022), Luo et al. (2023), Tirinzoni et al. (2025)) which can produce diverse behaviors trained using human demonstration data.

3 Preliminaries

We consider a reward-free Markov Decision Process (MDP) (Puterman, 2014) which is defined as a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, d_0, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ respectively denote the state and action spaces, P denotes the transition dynamics with $P(s'|s,a)$ indicating the probability of transitioning from s to s' by taking action a, d_0 denotes the initial state distribution and $\gamma \in (0,1)$ specifies the discount factor. A policy π is a function $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ mapping a state s to probabilities of action in $\mathcal{A}$. We denote by $\Pr(\cdot \mid s, a, \pi)$ and $\mathbb{E}[\cdot \mid s, a, \pi]$ the probability and expectation operators under state-action sequences $(s_t, a_t)_{t\geq 0}$ starting at (s,a) and following policy π with $s_t \sim P(\cdot \mid s_{t-1}, a_{t-1})$ and $a_t \sim \pi(\cdot \mid s_t)$. Given any reward function $r : \mathcal{S} \to \mathbb{R}$, the Q-function of π for r is $Q_r^\pi(s,a) := \sum_{t\geq 0} \gamma^t \mathbb{E}[r(s_{t+1}) \mid s, a, \pi]$.

Successor measures based Behavioral Foundation Models: The *successor measure* (Dayan, 1993; Blier et al., 2021) of state-action (s,a) under a policy π is the (discounted) distribution of future states obtained by taking action a in state s and following policy π thereafter:

$$M^\pi(s,a,X) := \sum_{t\geq 0} \gamma^t \Pr(s_{t+1} \in X \mid s, a, \pi) \quad \forall X \subset \mathcal{S}. \tag{1}$$

Q functions can be represented using successor measures as, $Q^\pi(s,a) = \sum_{s^+} M^\pi(s,a,s^+) r(s^+)$. This simple linear relationship between Q functions and Successor Measures have been exploited by a number of works (Touati & Ollivier, 2021; Agarwal et al., 2024) to create Behavioral Foundation Models(BFMs). The BFMs parameterize their policies (and correspondingly successor measures) using a latent $z \in \mathcal{Z}$ to pre-compute $\pi_z = \arg\max_z M_z^\top r$. BFMs based on successor features (Touati & Ollivier, 2021; Zheng et al., 2024) parameterize the reward functions linearly using these latents as spans of the state features, $\phi : \mathcal{S} \to \mathcal{Z}$, $r = \phi z$. Hence, the inference for any reward function reduces to finding this latent z from reward samples using linear regression. We will be following a similar setup as these successor feature methods, where we will be representing rewards as a span of the state features and learn to represent successor measures using these state features, leading to efficient computation of $M_z^\top r$.

4 Method

This method can be broadly divided into two parts - representation learning and zero-shot RL. The state representation encoder is trained using dynamics prediction and orthonormality loss, enabling the encoder to learn representations that will be generalizable across tasks. Leveraging these robust state embeddings, we then pretrain a Behavioral Foundation Model (BFM) to predict successor measures, enabling zero-shot inference of near-optimal policies for unseen reward functions. We refer to this method as `RLDP` (**R**egularized **L**atent **D**ynamics **P**rediction based Behavioral Foundation Policies)

4.1 Learning Representations with Regularized Latent Dynamics Prediction

Zero-shot RL based on successor measures rely on learning a state representation denoted by $\phi(s)$. This state representation will define the span of reward functions that the zero-shot RL method is guaranteed to output optimal policies for. Our primary representation learning objective is simple — unrolled latent dynamics prediction. We learn a state representation encoder $\phi : \mathcal{S} \to \mathbb{R}^d$, ($\mathcal{Z} = \mathbb{R}^d$) and a latent state-action representation encoder $g : \mathbb{R}^d \times \mathcal{A} \to \mathbb{R}^d$ such that latent dynamics representation remains linear in these representations $\phi(s') = g(\phi(s), a)^\top \mathbf{w}$ with some constant weights $\mathbf{w}$ informing our loss function for representation learning. A sub-sequence of horizon H is sampled from the offline interaction dataset given by $\tau^i = \{s^i_0, a^i_0, s^i_1, a^i_1, ..., s^i_{H-1}, a^i_{H-1}, s^i_H\}$. A sequence of future latent states $h^i_{1:H}$ are obtained by encoding the initial state $h^i_0 = \phi(s^i_0)$ and unrolling using the defined dynamics model $h^i_{t+1} = g(h^i_t, a_t)^\top.\mathbf{w}$. Then the objective is to predict the encoded future latent states:

$$\mathcal{L}_d = \mathbb{E}_{\tau^i \sim d^O} \left[\left\| \sum_{t=1}^{H} h^i_t - \bar{\phi}(s^i_t) \right\|^2 \right] \tag{2}$$

where $\bar{\phi}$ is slowly moving encoder target. The idea of learning low-rank representations for dynamics prediction is inspired by prior works in linear MDP, MDP homomorphisms and BYOL (Parr et al., 2008; Ravindran, 2004; Van der Pol et al., 2020; Grill et al., 2020) and has been shown to be successful in recent work of Fujimoto et al. (2025) where it is observed that model-free methods can be made competitive to model-based methods in sample efficiency and final performance with representations learned from dynamics prediction.

However, solely learning from the latent dynamics objective can lead to convergence to a collapsed solution. This is unsurprising as trivial solutions of predicting a constant zero vector achieves a perfect loss in Equation 2. To combat this, prior works (Grill et al., 2020) have proposed the use of a semi-gradient update where a stop-gradient is used for target h_{t+1} in Equation 2 along with a slowly updating target. Do these techniques mitigate solution collapse? We answer this question by plotting the cosine similarity of state representations trained via the above objective on an offline dataset collected by an exploration algorithm RND (Burda et al., 2019). Figure 1 shows that while the solutions do not collapse, there is an increase in feature similarity over the course of learning which we refer to as a mild form of collapse. As the space of reward functions is spanned by state features, such an increase in feature similarity can directly reduce the class of reward functions for which we can learn optimal policies.

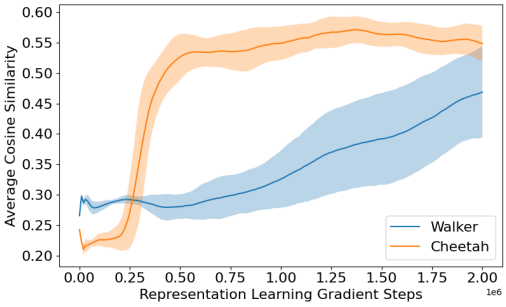

Figure 1: Average Cosine similarity between state-representations sampled uniformly from the training dataset: Feature similarity increases over the course of training. Shaded region shows standard deviation over 4 seeds

Preventing collapse in unsupervised RL: In the setting of unsupervised RL, the dataset contains purely reward-free transitions. To prevent collapse, we consider diversity regularization. Orthornormality regularizations have been widely studied in self-supervised learning (He et al., 2024b; Bansal et al., 2018b). Since we are looking to span reward functions using these state features, it makes sense to have these features orthogonal to each other. We project all state representations in a hypersphere: $\mathbb{S}^{d-1} = \{x \in \mathbb{R}^d \ : \ \|x\|_2 = 1\}$ and regularize by minimizing cosine similarity between any two states. This technique is also referred to as orthogonal regularization and has been used in self-supervised techniques for representation learning in vision and natural language (Bansal et al., 2018a; He et al., 2024a). Some prior unsupervised RL methods (Touati et al., 2023) use this regularization as an implementation trick to stabilize training; in the case of latent dynamics prediction this step becomes crucial to mitigate the increase in representation similarity. The orthogonal regularization loss looks like:

$$\mathcal{L}_r = \mathbb{E}_{s,s' \sim d^O}[\phi(s)^\top \phi(s')] \tag{3}$$

Our final loss is a weighted combination of dynamics prediction combined with orthogonal diversity regularization :

$$\mathcal{L} = \mathcal{L}_d + \lambda \mathcal{L}_r \tag{4}$$

where λ controls the regularization strength.

4.2 Zero-shot RL with Regularized Latent Dynamics Predictive Representations

We construct a Behavioral Foundation Model using the state-encoder ϕ trained by minimizing $\mathcal{L}$ (Equation 4). We define the class of reward functions as: $\mathcal{T} = \{\mathbf{r}_z | \mathbf{r}_z = (\phi^\top \phi)^{-1} \phi^\top z$ for some $z \in \mathbb{S}^{d-1}\}$ and define π_z to be the optimal policy for r_z. Successor Measures, M^{π_z}, are parameterized as $M^{\pi_z}(s, a, s^+) = \psi^{\pi_z}(s, a)\phi(s^+)$ with $\psi^{\pi_z}(s, a)$ (or $\psi(s, a, z)$) being the successor features for the state feature ϕ i.e. $\psi^{\pi_z}(s, a) = \mathbb{E}_{\pi_z}[\sum_{t=0}^{\infty} \gamma^t \phi(s)]$. Since ϕ is already obtained using Equation 4, learning M^{π_z} would mean optimizing for ψ. Following Touati & Ollivier (2021); Touati et al. (2023); Agarwal et al. (2024), we use a contrastive objective to train M^{π_z}, parameterized as $M^{\pi_z}(s, a, s^+) = \psi(s, a, z)\phi(s^+)$,

$$\begin{aligned}\mathcal{L}_{BFM} = &-\mathbb{E}_{s,a,s' \sim d^O}[M_z^\pi(s, a, s')] \\ &+ \frac{1}{2}\mathbb{E}_{s,a,s' \sim d^O, s^+ \sim d^O}[(M_z^\pi(s, a, s^+) - \gamma \bar{M}_z^\pi(s', \pi_z(s'), s^+))^2].\end{aligned} \tag{5}$$

Equation 5 requires samples from π_z. Hence the policy is optimized to maximize Q^{π_z},

$$\pi_z(s) = \max_a Q^{\pi_z}(s, a) = \max_a \psi(s, a, z).z \tag{6}$$

During inference, given a reward function r, we obtain the near-optimal policy by finding the corresponding z via linear regression,

$$z_R = \min_z \mathbb{E}_{d^O}[(r(s) - (\phi^\top \phi)^{-1}\phi(s) \cdot z)]^2 \implies z_R = \mathbb{E}_{d^O}[\phi(s) \cdot r(s)] \tag{7}$$

5 Experiments

The goal of our experiments is to perform an empirical study of suitability of state representations learned by latent next-state prediction objective when compared to other methods that employ more complex strategies. We perform several empirical ablations to understand our design choices. In particular, we aim to answer the following questions: a) Under an apples-to-apples setting of keeping all other learning factors similar, how does our method compare to baselines in enabling generalization to unseen reward functions? b) How does our method fare in the setting of large observation space of pixels? c) What design decisions are crucial to the success of our method?

Setup: We consider continuous control tasks from DeepMind control suite - Pointmass, Cheetah, Walker, Quadruped under a similar setup considered by prior works in zero-shot RL. A dataset is collected in these experiments using an exploration algorithm RND (Burda et al., 2019) without specifying any reward functions. With such data, we pretrain a BFM using the method outlined in

	Task	Laplace	FB	HILP	PSM	RLDP
Walker	Stand	243.70 ± 151.40	902.63 ± 38.94	607.07 ± 165.28	872.61 ± 38.81	890.40 ± 27.33
	Run	63.65 ± 31.02	392.76 ± 31.29	107.84 ± 34.24	351.50 ± 19.46	334.26 ± 49.69
	Walk	190.53 ± 168.45	877.10 ± 81.05	399.67 ± 39.31	891.44 ± 46.81	779.768 ± 137.156
	Flip	48.73 ± 17.66	206.22 ± 162.27	277.95 ± 59.63	640.75 ± 31.88	492.94 ± 22.79
	Average(*)	136.65	594.67	348.13	**689.07**	**624.34**
Cheetah	Run	96.32 ± 35.69	257.59 ± 58.51	68.22 ± 47.08	244.38 ± 80.00	157.12 ± 29.92
	Run Backward	106.38 ± 29.40	307.07 ± 14.91	37.99 ± 25.16	296.44 ± 20.14	170.52 ± 15.30
	Walk	409.15 ± 56.08	799.83 ± 67.51	318.30 ± 168.42	984.21 ± 0.49	592.92 ± 104.66
	Walk Backward	654.29 ± 219.81	980.76 ± 2.32	349.61 ± 236.29	979.01 ± 7.73	821.51 ± 50.62
	Average(*)	316.53	586.31	193.53	**626.01**	435.52
Quadruped	Stand	854.50 ± 41.47	740.05 ± 107.15	409.54 ± 97.59	842.86 ± 82.18	794.94 ± 43.25
	Run	412.98 ± 54.03	386.67 ± 32.53	205.44 ± 47.89	431.77 ± 44.69	457.41 ± 74.70
	Walk	494.56 ± 62.49	566.57 ± 53.22	218.54 ± 86.67	603.97 ± 73.67	465.40 ± 185.29
	Jump	642.84 ± 114.15	581.28 ± 107.38	325.51 ± 93.06	596.37 ± 94.23	733.322 ± 55.304
	Average(*)	601.22	568.64	289.75	**618.74**	**612.77**
Pointmass	Top Left	713.46 ± 58.90	897.83 ± 35.79	944.46 ± 12.94	831.43 ± 69.51	890.406 ± 60.791
	Top Right	581.14 ± 214.79	274.95 ± 197.90	96.04 ± 166.34	730.27 ± 58.10	795.469 ± 21.103
	Bottom Left	689.05 ± 37.08	517.23 ± 302.63	192.34 ± 177.48	451.38 ± 73.46	805.172 ± 20.443
	Bottom Right	21.29 ± 42.54	19.37 ± 33.54	0.17 ± 0.29	43.29 ± 38.40	193.381 ± 167.633
	Average(*)	501.23	427.34	308.25	514.09	**671.10**

Table 1: Comparison (over 4 seeds) of zero-shot RL performance between different methods. RLDP demonstrates a marked improvement over prior methods.

our method section 4. Each algorithm is given the same budget of gradient steps during pretraining, controlling the state representation dimension and the final performance is obtained by taking the model obtained at the end of learning and querying it for different task-rewards for 50 episodes. All of our results are aggregated across 4 seeds.

Baselines: We compare this method against commonly used state-of-the-art baselines in reward-function spanning/representation such as: FB, HILP, PSM, and Laplacian. The Laplacian approach (Wu et al., 2018) learns state representation using eigenvectors of graph-Laplacian induced by a random-walk operator. FB (Touati et al., 2023) intertwines learning of state-representation with policy learning step where state-representations are learned such that they can represent successor measures for a class of reward-optimal policies. HILP (Park et al., 2024) learns state representation using a goal reaching objective which is subsequently used for zero-shot RL. PSM (Agarwal et al., 2024) learns state representation to represent the successor measures for a class of policies defined with a discrete codebook. These baselines represent a set of diverse and strong approaches in the area of zero-shot RL.

Implementation and Evaluation: To evaluate the different zero-shot RL methods we take the pretrained policies and query them on a variety of tasks. For each environment, we consider 4 tasks similar to prior works (Touati et al., 2023; Park et al., 2024; Agarwal et al., 2024).

5.1 Benchmarking Zero-Shot RL for Continuous Control

We conduct our experiments across two axis - a) Table 1 compares against representation dimensions suggested by authors for different methods with the same number of gradient updates for pretraining each BFM. b) Table 2 pretrains all the BFMs on same number of representation dimensions (512) and gradient steps (2 million). We also conducted experiments for pixel-based inputs, the results for which are in Table 3. We consider four environments – Walker, Cheetah, Quadruped, and Pointmass and use the ExoRL suite (Yarats et al. (2022)) for obtaining exploratory datasets using RND (Burda et al. (2019)).

Overall, RLDP fares competitively to baselines across the environments despite its simplicity. Ablations studying the effects of the orthogonality loss and encoding horizon in representation learning are presented in the next section.

5.2 What matters for supervising representation suitable for control?

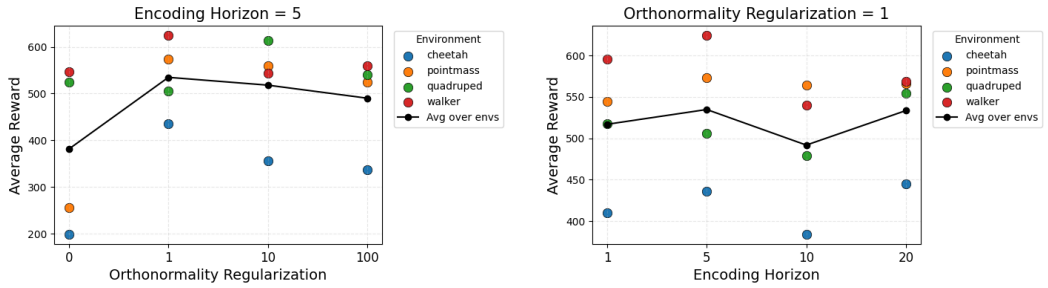

Figure 2: Evaluating the impact of Encoding Horizon and Orthogonality Regularization

In this section, we aim to ablate components of our method and understand which factors have a strong effect on final performance. To that end, we consider ablating diversity regularization as well as encoding horizon - the two design choices we make in our method. We begin by keeping the encoding horizon constant ($H = 5$) while we change the orthogonality regularization coefficient. For the second study, we keep the orthogonality regularizer constant ($\lambda = 1.0$) while changing the encoding horizon. The results for these ablations are in Figure 2.

	Task	FB	PSM	RLDP
Walker	Stand	918.29 ± 28.83	899.54 ± 30.73	890.40 ± 27.33
	Run	381.31 ± 17.32	450.57 ± 28.95	334.26 ± 49.69
	Walk	779.29 ± 63.60	875.61 ± 33.44	779.768 ± 137.16
	Flip	977.08 ± 2.76	621.36 ± 75.62	492.94 ± 22.79
	Average(*)	**763.99**	711.77	624.34
Cheetah	Run	129.39 ± 37.63	181.85 ± 54.17	157.12 ± 29.92
	Run Backward	142.41 ± 36.77	158.64 ± 18.56	170.52 ± 15.30
	Walk	604.54 ± 80.51	576.98 ± 209.45	592.92 ± 104.66
	Walk Backward	630.40 ± 144.23	817.92 ± 98.86	821.51 ± 50.62
	Average(*)	376.69	**433.85**	**435.52**
Quadruped	Stand	732.59 ± 101.33	708.03 ± 34.99	794.94 ± 43.25
	Run	425.15 ± 52.02	404.32 ± 23.26	457.41 ± 74.70
	Walk	492.91 ± 17.55	523.94 ± 52.13	465.40 ± 185.29
	Jump	567.27 ± 48.90	549.57 ± 15.86	733.322 ± 55.30
	Average(*)	554.48	546.46	**612.77**
Pointmass	Top Left	943.85 ± 17.31	924.20 ± 10.64	890.41 ± 60.79
	Top Right	550.84 ± 282.41	666.00 ± 133.15	795.47 ± 21.10
	Bottom Left	672.28 ± 153.06	800.93 ± 15.62	805.17 ± 20.44
	Bottom Right	272.97 ± 274.99	123.44 ± 138.82	193.38 ± 167.63
	Average(*)	461.77	488.05	**671.11**

Table 2: Comparison (over 4 seeds) of zero-shot RL performance between FB, PSM, and RLDP with representation size of $d = 512$. RLDP results are unchanged.

We observe that for zero regularization, the performance takes a steep dip compared to $\lambda > 0$. This shows that orthogonality regularizer is critical to the representation learning part of our algorithm. For fixed encoding horizon, we see that orthogonality regularizer $\lambda = 1$ performs best. To further understand the role of the orthogonality regularizer in representation learning and how it helps prevent feature collapse, we refer to Section 4.1 and Section 6.2, where we look at the cosine similarity between representations. For the second part of our ablations where we see how different encoding horizons can affect the average reward obtained by the agent under constant orthonormality regularization ($\lambda = 1$), we observe that encoding horizons 5 and 20 perform better than the others, with 5 performing best on average across all environments. For the purpose of our experiments, we use encoding horizon 5.

6 Conclusion

This paper introduces RLDP, a method to decouple representation learning and reinforcement learning, which allows for learning generalizable representations. Our objective takes the simple form of regularized latent-dynamics prediction, an objective that does not require any reconstruction, making it able to handle high-dimensional observation space and does not require Bellman backups, making it more amenable to optimization. We show that simply using latent-dynamics prediction

leads to a mild form of feature collapse where the state-representation similarity increases over time. To combat this issue, we propose using orthogonal regularization, a well-known technique to prevent feature collapse. Using our method enables learning generalizable, stable, and robust representations that can achieve competitive performance compared to established successor measure-based techniques without relying on reinforcement-driven signals. In this work, we present initial investigation results. Future research directions include qualitatively examining the learned representations, alternative regularization strategies, further scaling these methods to complex pixel-based observations, and extending the applicability to real-world robotics and control tasks. This work, thus, paves the way for simpler yet effective approaches to developing behavioral foundation models.

References

Joshua Achiam, Harrison Edwards, Dario Amodei, and Pieter Abbeel. Variational option discovery algorithms. *arXiv preprint arXiv:1807.10299*, 2018.

Siddhant Agarwal, Aaron Courville, and Rishabh Agarwal. Behavior predictive representations for generalization in reinforcement learning. In *Deep RL Workshop NeurIPS 2021*, 2021. URL https://openreview.net/forum?id=b5PJaxS6Jxg.

Siddhant Agarwal, Harshit Sikchi, Peter Stone, and Amy Zhang. Proto successor measure: Representing the space of all possible solutions of reinforcement learning. *arXiv preprint arXiv:2411.19418*, 2024.

Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. Can we gain more from orthogonality regularizations in training deep networks? *Advances in Neural Information Processing Systems*, 31, 2018a.

Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. Can we gain more from orthogonality regularizations in training deep cnns? *CoRR*, abs/1810.09102, 2018b. URL http://arxiv.org/abs/1810.09102.

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

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.

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

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, Ruslan Salakhutdinov, and Sergey Levine. The information geometry of unsupervised reinforcement learning. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=3wU2UX0voE.

Scott Fujimoto, Pierluca D'Oro, Amy Zhang, Yuandong Tian, and Michael Rabbat. Towards general-purpose model-free reinforcement learning. *arXiv preprint arXiv:2501.16142*, 2025.

Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18995–19012, 2022.

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.

Junlin He, Jinxiao Du, and Wei Ma. Preventing dimensional collapse in self-supervised learning via orthogonality regularization. *arXiv preprint arXiv:2411.00392*, 2024a.

Junlin He, Jinxiao Du, and Wei Ma. Preventing dimensional collapse in self-supervised learning via orthogonality regularization, 2024b. URL https://arxiv.org/abs/2411.00392.

Aviral Kumar, Rishabh Agarwal, Tengyu Ma, Aaron Courville, George Tucker, and Sergey Levine. Dr3: Value-based deep reinforcement learning requires explicit regularization. *arXiv preprint arXiv:2112.04716*, 2021.

Zhengyi Luo, Jinkun Cao, Josh Merel, Alexander Winkler, Jing Huang, Kris Kitani, and Weipeng Xu. Universal humanoid motion representations for physics-based control. *arXiv preprint arXiv:2310.04582*, 2023.

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.

Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *6th Annual Conference on Robot Learning*.

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

Seohong Park, Tobias Kreiman, and Sergey Levine. Foundation policies with hilbert representations. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=LhNsSaAKub.

Ronald Parr, Lihong Li, Gavin Taylor, Christopher Painter-Wakefield, and Michael L Littman. An analysis of linear models, linear value-function approximation, and feature selection for reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pp. 752–759, 2008.

Xue Bin Peng, Yunrong Guo, Lina Halper, Sergey Levine, and Sanja Fidler. Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters. *ACM Transactions On Graphics (TOG)*, 41(4):1–17, 2022.

Matteo Pirotta, Andrea Tirinzoni, Ahmed Touati, Alessandro Lazaric, and Yann Ollivier. Fast imitation via behavior foundation models. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*, 2023.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *CoRR*, abs/2102.12092, 2021. URL https://arxiv.org/abs/2102.12092.

Balaraman Ravindran. *An algebraic approach to abstraction in reinforcement learning*. University of Massachusetts Amherst, 2004.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations. *arXiv preprint arXiv:2007.05929*, 2020.

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. *arXiv preprint arXiv:2504.07896*, 2025.

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. *arXiv preprint arXiv:2504.11054*, 2025.

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.

Elise Van der Pol, Thomas Kipf, Frans A Oliehoek, and Max Welling. Plannable approximations to mdp homomorphisms: Equivariance under actions. *arXiv preprint arXiv:2002.11963*, 2020.

David Warde-Farley, Tom Van de Wiele, Tejas Kulkarni, Catalin Ionescu, Steven Hansen, and Volodymyr Mnih. Unsupervised control through non-parametric discriminative rewards. *arXiv preprint arXiv:1811.11359*, 2018.

Jungdam Won, Deepak Gopinath, and Jessica Hodgins. Physics-based character controllers using conditional vaes. *ACM Transactions on Graphics (TOG)*, 41(4):1–12, 2022.

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

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

Supplementary Materials

The following content was not necessarily subject to peer review.

Appendix

6.1 Pseudocode for the update

Algorithm 1 UPDATE RLDP (PRETRAINING)

Require: Replay buffer $\mathcal{D}$; encoder ϕ with target $\bar{\phi}$; forward model ψ with target $\bar{\psi}$; actor π
Require: Pretraining steps T_{repr}, Polyak factor τ
  for num_train_steps **do**
    if step $\leq$ representation_steps **then**
      Representation Learning (Section 4.1):
      Sample segment batch $\{s_{0:H}, a_{0:H}\} \sim \mathcal{D}$
      $L_{\text{repr}} \leftarrow \lambda\, L_{\text{orth}} + L_{\text{dyn}}$ (Equation 2, 3, 4)
      if step$\%250 == 0$ **then**
        $\bar{\phi} \leftarrow \phi$
      end if
    else
      Sample transitions $\{(s, a, s', \text{done})\} \sim \mathcal{D}$
      $\gamma \leftarrow \gamma_0\,(1 - \text{done})$
      Sample $z \sim \text{MixUniform} \cup \phi(s')$ {mix random prior + goal-encoded}
      Value update (Section 4.2):
      $a' \sim \pi(s', z)$
      $F^\star \leftarrow \bar{\psi}(s', z, a'), \quad B^\star \leftarrow \bar{\phi}(s')$
      $M^\star_{ij} = F^\star_i \cdot B^\star_j$
      $F \leftarrow \psi(s, z, a), \quad B \leftarrow \phi(s')$
      $M_{ij} = F_i \cdot B_j$
      $L_{FB} = \frac{1}{2}\,\mathbb{E}[\|(M - \gamma M^\star) \circ \mathbf{1}_{i \neq j}\|^2]$
      Update $\psi \leftarrow \psi - \eta \nabla_\psi L_{FB}$
      Policy update (Section 4.2):
      $a = \pi(s, z)$
      $Q = \psi(s, z, a) \cdot z$
      $L_\pi = -\mathbb{E}[Q]$
      Update $\pi \leftarrow \pi - \eta \nabla_\pi L_\pi$
      Target-network sync:
      $\bar{\psi} \leftarrow \tau\psi + (1 - \tau)\,\bar{\psi}$
    end if
  end for

6.2 Additional Results

Learning Zero-shot Policies for Continuous Control Table 3 shows the performance of RLDP on pixel inputs.

Understanding the role of orthogonality regularization Figure 3 shows the impact of changing orthogonality regularization while keeping a constant encoding horizon ($H = 5$). Specifically, we look at how the cosine similarity between representations changes during training for different regularization coefficients.

We observe that for a regularization coefficient of 0, the cosine similarity increases, indicating that all states are getting mapped to similar representations, i.e., representation collapse. For any regularization coefficient greater than 0 ($\lambda > 0$), we observe that the cosine similarity drops below 0.1,

Dataset	Environment	Task	FDM	FB	HILP	RLDP
RND	Walker	Flip	282 ± 52	62 ± 57	232 ± 41	242.6 ± 33.5
		Run	146 ± 60	42 ± 25	126 ± 8	106.6 ± 12.3
		Stand	557 ± 97	172 ± 111	496 ± 73	464.1 ± 67.7
		Walk	452 ± 52	104 ± 82	376 ± 52	229.8 ± 55.4
	Average(*)		**359.25**	95.0	307.5	149.0
	Cheetah	Run	178 ± 41	221 ± 15	276 ± 46	101.1 ± 30.3
		Run Backward	126 ± 9	171 ± 123	297 ± 46	31.4 ± 5.5
		Walk	470 ± 182	535 ± 251	895 ± 33	439.7 ± 133.7
		Walk Backward	441 ± 107	535 ± 440	927 ± 35	137.1 ± 27.2
	Average(*)		303.8	365.5	**598.8**	177.3
	Quadruped	Jump	273 ± 66	224 ± 149	244 ± 122	430.5 ± 94.9
		Run	192 ± 29	158 ± 82	148 ± 19	323.4 ± 37.1
		Stand	374 ± 85	347 ± 191	327 ± 126	507.6 ± 76.8
		Walk	199 ± 63	162 ± 92	163 ± 45	285.7 ± 29.9
	Average(*)		259.5	222.8	220.5	**386.8**

Table 3: Performance comparison on the pixel-based ExORL benchmark across different environments and tasks.

close to 0.0, indicating that the states are being mapped to different representations. This highlights the importance of the regularization coefficient in preventing representation collapse.

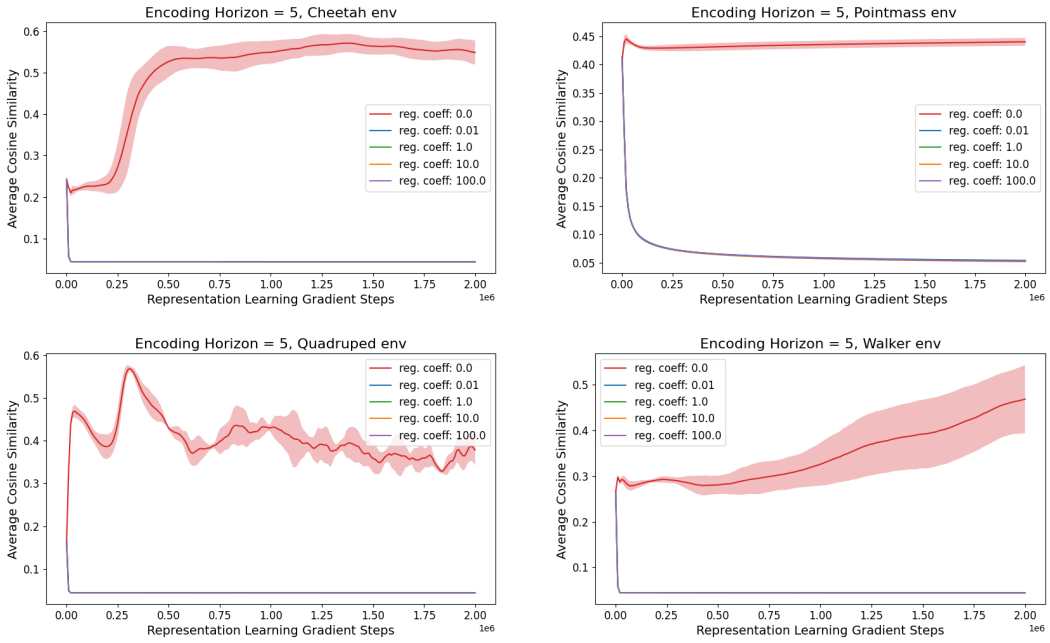

Figure 3: Evaluating the impact of Orthogonality Regularization on representations learned across four environments: Cheetah (top left), Pointmass (top right), Quadruped (bottom left), and Walker (bottom right).