BRIDGING STATE AND HISTORY REPRESENTATIONS: UNDERSTANDING SELF-PREDICTIVE RL

Tianwei Ni[♣], **Benjamin Eysenbach**[♠], **Erfan Seyedsalehi**[◇*], **Michel Ma**[♣], **Clement Gehring**[♣], **Aditya Mahajan**[◇], **Pierre-Luc Bacon**[♣]
[♣]Mila, Université de Montréal, [♠]Princeton University, [◇]Mila, McGill University
{tianwei.ni, michel.ma, clement.gehring, pierre-luc.bacon}@mila.quebec,
eysenbach@princeton.edu, erfan.seyedsalehi@mail.mcgill.ca, aditya.mahajan@mcgill.ca

ABSTRACT

Representations are at the core of all *deep* reinforcement learning (RL) methods for both Markov decision processes (MDPs) and partially observable Markov decision processes (POMDPs). Many representation learning methods and theoretical frameworks have been developed to understand what constitutes an effective representation. However, the relationships between these methods and the shared properties among them remain unclear. In this paper, we show that many of these seemingly distinct methods and frameworks for state and history abstractions are, in fact, based on a common idea of *self-predictive* abstraction. Furthermore, we provide theoretical insights into the widely adopted objectives and optimization, such as the stop-gradient technique, in learning self-predictive representations. These findings together yield a minimalist algorithm to learn self-predictive representations for states and histories. We validate our theories by applying our algorithm to standard MDPs, MDPs with distractors, and POMDPs with sparse rewards. These findings culminate in a set of preliminary guidelines for RL practitioners.[1]

1 INTRODUCTION

Reinforcement learning holds great potential to automatically learn optimal policies, mapping observations to return-maximizing actions. However, the application of RL in the real world encounters challenges when observations are high-dimensional and/or noisy. These challenges become even more severe in partially observable environments, where the observation (history) dimension grows over time. In fact, current RL algorithms are often brittle and sample inefficient in these settings (Wang et al., 2019; Stone et al., 2021; Tomar et al., 2021; Morad et al., 2023).

To address the curse of dimensionality, a substantial body of work has focused on compressing observations into a latent state space, known as state abstraction in MDPs (Dayan, 1993; Dean & Givan, 1997; Li et al., 2006), history abstraction in POMDPs (Littman et al., 2001; Castro et al., 2009), and sufficient statistics or information states in stochastic control (Striebel, 1965; Kwakernaak, 1965; Bohlin, 1970; Kumar & Varaiya, 1986). Traditionally, this compression has been achieved through hand-crafted feature extractors (Sutton, 1995; Konidaris et al., 2011) or with the discovery of a set of core tests sufficient for predicting future observations (Littman et al., 2001; Singh et al., 2003). Modern approaches learn the latent state space using an encoder to automatically filter out irrelevant parts of observations (Lange & Riedmiller, 2010; Watter et al., 2015; Munk et al., 2016). Furthermore, *deep* RL enables end-to-end and online learning of compact state or history representations alongside policy training. As a result, numerous representation learning techniques for RL have surfaced (refer to Table 1), drawing inspiration from diverse fields within ML and RL. However, this abundance of methods may have inadvertently presented practitioners with a "paradox of choice", hindering their ability to identify the best approach for their specific RL problem.

This paper aims to offer systematic guidance regarding the essential characteristics that good representations should possess in RL (the "**what**") and effective strategies for learning such representations (the "**how**"). We begin our analysis from first principles by comparing and connecting various representations proposed in prior works for MDPs and POMDPs, resulting in a unified view. Remarkably, these representations are all connected by a **self-predictive** condition – the encoder can predict its next latent state (Subramanian et al., 2022). Next, we examine how to learn such self-predictive condition

[*]Work done while ES was at McGill University.
[1]Please refer to https://arxiv.org/abs/2401.08898 for the 10-main-page version of this paper.

in RL, a difficult subtask due to the bootstrapping effect (Gelada et al., 2019; Schwarzer et al., 2020; Tang et al., 2022). We provide fresh insights on why the popular "stop-gradient" technique, in which the parameters of the encoder do not update when used as a target, has the promise of learning the desired condition without representational collapse in POMDPs. Building on our new theoretical findings, we introduce a minimalist RL algorithm that learns self-predictive representations end-to-end with a *single* auxiliary loss, *without* the need for reward model learning (thereby removing planning) (François-Lavet et al., 2019; Gelada et al., 2019; Tomar et al., 2021; Hansen et al., 2022; Ghugare et al., 2022; Ye et al., 2021; Subramanian et al., 2022), reward regularization (Eysenbach et al., 2021), multi-step predictions and projections (Schwarzer et al., 2020; Guo et al., 2020), and metric learning (Zhang et al., 2020; Castro et al., 2021). Furthermore, the simplicity of our approach allows us to investigate the role of representation learning in RL, in isolation from policy optimization.

The core contributions of this paper are as follows. We establish a unified view of state and history representations with novel connections (Sec. 3), revealing that many prior methods optimize a collection of closely interconnected properties, each representing a different facet of the same fundamental concept. Moreover, we enhance the understanding of self-predictive learning in RL regarding the choice of the objective and its impact on the optimization dynamics (Sec. 4). Our theory results in a simplified and novel RL algorithm designed to learn self-predictive representations fully end-to-end (Sec. 4.3). Through extensive experimentation across three benchmarks (Sec. 5), we provide empirical evidence substantiating all our theoretical predictions using our simple algorithm. Finally, we offer our recommendations for RL practitioners in Sec. 6. Taken together, we believe that our work potentially aids in addressing the longstanding challenge of learning representations in MDPs and POMDPs.

2 Background

MDPs and POMDPs. In the context of a POMDP $\mathcal{M}_O = (\mathcal{O}, \mathcal{A}, P, R, \gamma, T)$[2], an agent receives an observation $o_t \in \mathcal{O}$ at time step t, selects an action $a_t \in \mathcal{A}$ based on the observed history $h_t := (h_{t-1}, a_{t-1}, o_t) \in \mathcal{H}_t$[3], and obtains a reward $r_t \sim R(h_t, a_t)$ along with the subsequent observation $o_{t+1} \sim P(\cdot \mid h_t, a_t)$. The initial observation $h_1 := o_1$ is sampled from the distribution $P(o_1)$. The total time horizon is denoted as $T \in \mathbb{N}^+ \cup \{+\infty\}$, and the discount factor is $\gamma \in [0, 1]$ (less than 1 for infinite horizon). To maintain brevity, we employ the "prime" symbol to represent the next time step, for example writing $h' = (h, a, o')$. Under the above assumptions, our agent acts according to a policy $\pi(a \mid h)$ with action-value $Q^\pi(h, a)$. Furthermore, it can be shown that there exists an optimal value function $Q^*(h, a)$ such that $Q^*(h, a) = \mathbb{E}[r \mid h, a] + \gamma \mathbb{E}_{o' \sim P(|h,a)}[\max_{a'} Q^*(h', a')]$, and a deterministic optimal policy $\pi^*(h) = \operatorname{argmax}_a Q^*(h, a)$. In an MDP $\mathcal{M}_S = (\mathcal{S}, \mathcal{A}, P, R, \gamma, T)$, the observation o_t and history h_t are replaced by the state $s_t \in \mathcal{S}$[4].

State and history representations. In a POMDP, an **encoder** is a function $\phi : \mathcal{H}_t \to \mathcal{Z}$ that produces a history **representation** $z = \phi(h) \in \mathcal{Z}$. Similarly, in an MDP, we replace h with s, resulting in a state encoder $\phi : \mathcal{S} \to \mathcal{Z}$ and a state representation $z = \phi(s) \in \mathcal{Z}$. This representation is known as an "abstraction" (Li et al., 2006) or a "latent state" (Gelada et al., 2019). Such encoders are sometimes shared and simultaneously updated by downstream components (*e.g.* policy, value, world model) of an RL system (Hafner et al., 2020a; Hansen et al., 2022). In this paper, we are interested in such a shared encoder, or **the encoder learned for the value function** if the encoders are separately learned.

Below, we present the key abstractions that are central to this paper, along with their established connections. We will highlight the **conditions** met by each abstraction. We defer additional common abstractions and related concepts to Sec. A.2.

1. Q^*-irrelevance abstraction. An encoder ϕ_{Q^*} provides a Q^*-irrelevance abstraction (Li et al., 2006) if it contains the necessary information for predicting the return. Formally, if $\phi_{Q^*}(h_i) = \phi_{Q^*}(h_j)$, then $Q^*(h_i, a) = Q^*(h_j, a), \forall a$. A Q^*-irrelevance abstraction can be achieved as a by-product of learning an encoder ϕ through a value function $\mathcal{Q}(\phi(h), a)$ end-to-end using model-free RL. If the optimal values match, then $\mathcal{Q}^*(\phi_{Q^*}(h), a) = Q^*(h, a), \forall h, a$.

2. Self-predictive (model-irrelevance) abstraction. We view the model-irrelevance concept (Li et al., 2006) from a self-predictive standpoint. Specifically, a model-irrelevant encoder ϕ_L fulfills two

[2]While the classic definition of a POMDP (Cassandra et al., 1994) features a state space, we assume it to be unknown, thus our view of a POMDP is a black-box input-output system (Subramanian et al., 2022).

[3]In general, $h_t := (h_{t-1}, a_{t-1}, r_{t-1}, o_t)$ (Izadi & Precup, 2005). In this study, we assume rewards are inaccessible during policy inference.

[4]In finite-horizon MDPs, we assume s includes the time step t.

conditions: **expected reward prediction (RP)** and **next latent state distribution prediction (ZP)**[5], ensuring that the encoder can be used to predict expected reward and the next latent state distribution.

$$\exists R_z : \mathcal{Z} \times \mathcal{A} \to \mathbb{R}, \quad s.t. \quad \mathbb{E}[r \mid h, a] = R_z(\phi_L(h), a), \quad \forall h, a, \tag{RP}$$

$$\exists P_z : \mathcal{Z} \times \mathcal{A} \to \Delta(\mathcal{Z}), \quad s.t. \quad P(z' \mid h, a) = P_z(z' \mid \phi_L(h), a), \quad \forall h, a, z', \tag{ZP}$$

$$\mathbb{E}[z' \mid h, a] = \mathbb{E}[z' \mid \phi_L(h), a], \quad \forall h, a. \tag{EZP}$$

A weak version of ZP is the **expected next latent state z prediction (EZP)** condition. ZP can be interpreted as a sufficient statistics condition on ϕ_L: the next latent state z' is conditionally independent of the history h when $\phi_L(h)$ and a is known, symbolized as $z' \perp\!\!\!\perp h \mid \phi_L(h), a$. Satisfying ZP only is trivial and can be achieved by employing a constant representation $\phi(h) = c$, where c is a fixed constant. Therefore, ZP must be used in conjunction with other conditions (*e.g.*, RP) to avoid such degeneration. The ϕ_L is known as a bisimulation generator (Givan et al., 2003) in MDPs and an information state generator (Subramanian et al., 2022) in POMDPs.

3. Observation-predictive (belief) abstraction. This abstraction is implicitly introduced by Subramanian et al. (2022), which we denote by ϕ_O, and satisfies three conditions: expected reward prediction RP, **recurrent encoder (Rec)** and **next observation distribution prediction (OP)**[6].

$$\exists \psi_z : \mathcal{Z} \times \mathcal{A} \times \mathcal{O} \to \mathcal{Z}, \quad s.t. \quad \phi(h') = \psi_z(\phi_O(h), a, o'), \quad \forall h, a, o', \tag{Rec}$$

$$\exists P_o : \mathcal{Z} \times \mathcal{A} \to \Delta(\mathcal{O}), \quad s.t. \quad P(o' \mid h, a) = P_o(o' \mid \phi_O(h), a), \quad \forall h, a, o', \tag{OP}$$

$$\exists \psi_o : \mathcal{Z} \to \mathcal{O}, \quad s.t. \quad o = \psi_o(\phi_O(h)), \quad \forall h. \tag{OR}$$

Similarly, the OP condition is equivalent to $o' \perp\!\!\!\perp h \mid \phi_O(h), a$, and OP is closely related to **observation reconstruction (OR)**, widely used in practice (Yarats et al., 2021). The recurrent encoder (Rec) condition is satisfied for encoders parameterized with feedforward or recurrent neural networks (Elman, 1990; Hochreiter & Schmidhuber, 1997), but not Transformers (Vaswani et al., 2017). In this paper, we assume the Rec condition is always satisfied. In POMDPs, ϕ_O is well-known as a belief state generator (Kaelbling et al., 1998).

We extend the relations between these abstractions known in MDPs (Li et al., 2006) to POMDPs.

Theorem 1 (**Relationships between common abstractions (informal)**). *An encoder satisfying ϕ_O also belongs to ϕ_L; an encoder satisfying ϕ_L also belongs to ϕ_{Q^*}; the reverse is not necessarily true.*

3 A Unified View on State and History Representations

3.1 An Implication Graph of Representations in RL

Using the taxonomy of state and history abstractions, it becomes possible to establish theoretical links among different representations and their respective conditions discussed earlier. These connections are succinctly illustrated in a directed graph, as shown in Fig. 1. In this section, we highlight the most significantly novel finding, while postponing the other propositions and proofs to Sec. A.

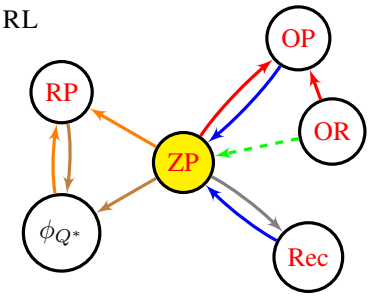

Figure 1: **An implication graph** showing the relations between the conditions on history representations. The source nodes of the edges with the same color *together* imply the target node. The dashed edge means it only applies to MDPs. All the connections are discovered in this work, except for (1) OP + Rec implying ZP, (2) ZP + RP implying ϕ_{Q^*}.

The definition of self-predictive and observation-predictive abstractions suggests the classic *phased* training framework. In phased training, we alternatively train an encoder to predict expected rewards (RP) and predict next latent states (ZP) or next observations (OP), and also train an RL or planning agent on the latent space with the encoder "detached" from downstream components. On the other hand, we show in our Thm. 2 that if we learn an encoder *end-to-end* in a model-free fashion but using ZP (or OP) as an auxiliary task, then the ground-truth expected reward can be induced by the latent Q-value and latent transition. Thus, the encoder also satisfies RP and generates ϕ_L (or ϕ_O) representation already.

Theorem 2 (**ZP + ϕ_{Q^*} imply RP**). *If an encoder ϕ satisfies ZP, and $\mathcal{Q}(\phi(h), a) = Q^*(h, a), \forall h, a$, then we can construct a latent reward function $\mathcal{R}_z(z, a) := \mathcal{Q}(z, a) - \gamma \mathbb{E}_{z' \sim P_z(|z,a)}[\max_{a'} \mathcal{Q}(z', a')]$, such that $\mathcal{R}_z(\phi(h), a) = \mathbb{E}[r \mid h, a], \forall h, a$.*

[5] RP and ZP are labeled as (P1) and (P2), respectively, in Subramanian et al. (2022).

[6] OP and Rec are labeled as (P2a) and (P2b), respectively, in Subramanian et al. (2022).

Table 1: **Which optimal representation will be learned by the value function in prior works?** The "PO" column shows if the approach applies to POMDPs. The "Conditions" column shows the conditions that the encoder of the optimal value satisfies (see the appendix for the "metric" and "regularization" conditions). The ZP loss shows the loss function they use to learn ZP condition. The ZP target shows whether they use online or stop-gradient (including detached and EMA) encoder target. Due to the space limit, we omit the citations for recurrent model-free RL (Hausknecht & Stone, 2015; Kapturowski et al., 2018; Ni et al., 2022) and belief-based methods (Wayne et al., 2018; Hafner et al., 2019; Han et al., 2020; Lee et al., 2020).

Work	PO?	Abstraction	Conditions	ZP loss	ZP target
Model-Free & Classic Model-Based RL	✗	ϕ_{Q^*}	ϕ_{Q^*}	N/A	N/A
MuZero (Schrittwieser et al., 2020)	✗	unknown	ϕ_{Q^*} + RP	N/A	N/A
MICo (Castro et al., 2021)	✗	unknown	ϕ_{Q^*} + metric	N/A	N/A
CRAR (François-Lavet et al., 2019)	✗	ϕ_L	ϕ_{Q^*} + RP + ZP + reg.	ℓ_2	online
DeepMDP (Gelada et al., 2019)	✗	ϕ_L	ϕ_{Q^*} + RP + ZP	W (ℓ_2)	online
SPR (Schwarzer et al., 2020)	✗	ϕ_L	ϕ_{Q^*} + ZP	cos	EMA
DBC (Zhang et al., 2020)	✗	ϕ_L	ϕ_{Q^*} + RP + ZP + metric	FKL	detached
LSFM (Lehnert & Littman, 2020)	✗	ϕ_L	ϕ_{Q^*} + RP + EZP	SF	detached
Baseline in (Tomar et al., 2021)	✗	ϕ_L	ϕ_{Q^*} + RP + ZP	ℓ_2	detached
EfficientZero (Ye et al., 2021)	✗	ϕ_L	ϕ_{Q^*} + RP + ZP	cos	detached
TD-MPC (Hansen et al., 2022)	✗	ϕ_L	ϕ_{Q^*} + RP + ZP	ℓ_2	EMA
ALM (Ghugare et al., 2022)	✗	ϕ_L	ϕ_{Q^*} + ZP	RKL	EMA
TCRL (Zhao et al., 2023)	✗	ϕ_L	RP + ZP	cos	EMA
OFENet (Ota et al., 2020)	✗	ϕ_O	ϕ_{Q^*} + OP	N/A	N/A
Recurrent Model-Free RL	✓	ϕ_{Q^*}	ϕ_{Q^*}	N/A	N/A
PBL (Guo et al., 2020)	✓	ϕ_L	ϕ_{Q^*} + ZP	ℓ_2	detached
AIS (Subramanian et al., 2022)	✓	ϕ_L, ϕ_O	RP + ZP or OP	ℓ_2, FKL	detached
Belief-Based Methods	✓	ϕ_O	RP + ZP + OR	FKL	online
Causal States (Zhang et al., 2019)	✓	ϕ_O	RP + OP	N/A	N/A
Minimalist ϕ_L (this work)	✓	ϕ_L	ϕ_{Q^*} + ZP	ℓ_2, KL	stop-grad

This result holds significance, as it suggests that end-to-end approaches to learning ϕ_{Q^*} + ZP (OP) have similar theoretical justification as classic phased approaches.

3.2 Which Representations Do Prior Methods Learn?

With the unified view of state and history representations, we can categorize prior works based on the conditions satisfied by the *optimal* encoders of their value functions. Table 1 shows representative examples. The unified view enables us to draw interesting connections between prior works, even though they may differ in RL or planning algorithms and the encoder objectives. Here we highlight some important connections and provide a more detailed discussion of all prior works in Sec. C.

To begin with, it is important to recognize that classic model-based RL actually learns ϕ_{Q^*} in value function. Model-based RL trains a policy and value by rolling out on the learned model. However, the policy and value do not share representations with the model (Sutton, 1990; Sutton et al., 2012; Chua et al., 2018; Kaiser et al., 2019; Janner et al., 2019), or learn their representations from maximizing returns (Tamar et al., 2016; Oh et al., 2017; Silver et al., 2017). Secondly, as shown in Table 1, there is a wealth of prior work on approximating ϕ_L, stemming from different perspectives. These include bisimulation (Gelada et al., 2019), information states (Subramanian et al., 2022), variational inference (Eysenbach et al., 2021; Ghugare et al., 2022), successor features (Barreto et al., 2017; Lehnert & Littman, 2020), and self-supervised learning (Schwarzer et al., 2020; Guo et al., 2020). The primary differences between these approaches lie in their selection of (1) architecture (whether learning RP, ϕ_{Q^*}, or both), (2) ZP objectives (such as ℓ_2, cosine, forward or reverse KL, as discussed in Sec. 4.1), and (3) ZP targets for optimization (including online, detached, EMA, as detailed in Sec. 4.2). Finally, observation-predictive representations are typically studied in POMDPs, where they are known as belief states (Kaelbling et al., 1998) and predictive state representations (Littman et al., 2001).

4 On Learning Self-Predictive Representations in RL

The implication graph (Fig. 1) establishes the theoretical connections among various representations in RL, yet it does not address the core learning problems. This section aims to give some theoretical answers to **how** to learn self-predictive representations. While self-predictive representation holds promise, it poses significant learning challenges compared to grounded model-free and observation-predictive representations. The bootstrapping effect, where ϕ appears in both sides of ZP (since z' also relies on $\phi(h')$), contributes to this difficulty. We present detailed analyses of the objectives in Sec. 4.1 and optimization in Sec. 4.2, with proofs deferred to Sec. B. Building on these analyses, we propose a simple representation learning algorithm for ϕ_L in RL in Sec. 4.3.

4.1 ARE PRACTICAL ZP OBJECTIVES BIASED?

Thm. 2 suggests that we can learn ϕ_L by simply training an auxiliary task of ZP on a model-free agent. Prior works have proposed several auxiliary losses, summarized in Table 1's ZP loss column. Formally, we parametrize an encoder with $f_\phi : \mathcal{H}_t \to \mathcal{Z}$ (deterministic case) or $f_\phi : \mathcal{H}_t \to \Delta(\mathcal{Z})$ (probabilistic case)[7]. The latent transition function $P_z(z' \mid z, a)$ is parameterized by $g_\theta : \mathcal{Z} \times \mathcal{A} \to \mathcal{Z}$ (deterministic case) or $g_\theta : \mathcal{Z} \times \mathcal{A} \to \Delta(\mathcal{Z})$ (probabilistic case). We use $\mathbb{P}_\phi(z \mid h)$ and $\mathbb{P}_\theta(z' \mid z, a)$ to represent the encoder and latent transition, respectively. The self-predictive metric for ZP is *ideally*:

$$\mathcal{L}_{ZP,\mathbb{D}}(\phi, \theta; h, a) := \mathbb{E}_{z \sim \mathbb{P}_\phi(|h)}[\mathbb{D}(\mathbb{P}_\theta(z' \mid z, a) \,||\, \mathbb{P}_\phi(z' \mid h, a))], \tag{1}$$

where $\mathbb{P}_\phi(z' \mid h, a) = \mathbb{E}_{o' \sim P(\cdot|h,a)}[\mathbb{P}_\phi(z' \mid h')]$. $\mathbb{D}(\cdot \,||\, \cdot) \in \mathbb{R}_{\geq 0}$ compares two distributions. When Eq. 1 reaches minimum, then for any $z \sim \mathbb{P}_\phi(| \, h)$, the ZP condition is satisfied.

When designing *practical* ZP loss, prior works are mainly divided into deterministic ℓ_2 approach (Gelada et al., 2019; Schwarzer et al., 2020; Tomar et al., 2021; Hansen et al., 2022; Ye et al., 2021)[8] or probabilistic `f`-divergence approach (Zhang et al., 2020; Ghugare et al., 2022; Hafner et al., 2019) that includes forward and reverse KL divergences (in short, FKL and RKL):

$$J_\ell(\phi, \theta, \widetilde{\phi}; h, a) := \mathbb{E}_{o' \sim P(|h,a)}\Big[\|g_\theta(f_\phi(h), a) - f_{\widetilde{\phi}}(h')\|_2^2\Big], \tag{2}$$

$$J_{D_\mathtt{f}}(\phi, \theta, \widetilde{\phi}; h, a) := \mathbb{E}_{z \sim \mathbb{P}_\phi(|h), o' \sim P(|h,a)}\Big[D_\mathtt{f}\left(\mathbb{P}_{\widetilde{\phi}}(z' \mid h') \,||\, \mathbb{P}_\theta(z' \mid z, a)\right)\Big], \tag{3}$$

where $\widetilde{\phi}$, called ZP **target**, can be **online** (exact ϕ that allows gradient backpropagation), or the **stop-gradient** version $\overline{\phi}$ (detached from the computation graph and using a copy or exponential moving average (EMA) of ϕ). The update rule is $\overline{\phi} \leftarrow \tau\overline{\phi} + (1-\tau)\phi$, with $\tau = 0$ for **detached** and $\tau \in (0, 1)$ for generic **EMA**. We summarize the choices of ZP targets in one column of Table 1.

We first investigate the relationship between the ideal objective Eq. 1 and practical objectives Eq. 2 and Eq. 3 to better understand their implications.

Proposition 1 (The practical ℓ_2 objective Eq. 2 is an **upper bound** of the ideal objective Eq. 1 $\mathcal{L}_{ZP,\ell}(\phi, \theta; h, a)$ that targets EZP condition. The equality holds in deterministic environments.)**.**

Proposition 2 (The practical `f`-divergence objective Eq. 3 is an **upper bound** of the ideal objective Eq. 1 $\mathcal{L}_{ZP,D_\mathtt{f}}(\phi, \theta; h, a)$ that targets ZP condition. The equality holds in deterministic environments.)**.**

These propositions show that environment stochasticity ($P(o' \mid h, a)$) affects both the practical ℓ_2 and `f`-divergence objectives. While unbiased in *deterministic* tasks to learn the ZP condition[9], they are problematic in *stochastic* tasks (*e.g.* with data augmentation or noisy distractors) due to double sampling issue (Baird, 1995), as the ideal objective Eq. 1 cannot be used (see Sec. B for discussion).

4.2 WHY DO STOP-GRADIENTS WORK FOR ZP OPTIMIZATION?

In this subsection, we further discuss optimizing the practical ZP objective Eq. 2. Specifically, we aim to justify that stop-gradient (detached or EMA) ZP targets, widely used in practice (Schwarzer et al., 2020; Zhang et al., 2020; Ghugare et al., 2022), play an important role in optimization. We find that they may lead to EZP condition in stochastic environments (Prop. 3) and can avoid representational collapse under some linear assumptions (Thm. 3). Meanwhile, online ZP targets lack these properties.

Before introducing the results, we want to clarify the discrepancy between learning ZP and the well-known TD learning (Sutton, 1988). At first glance, Eq. 2 (stop-gradient version) is reminiscent of mean-squared TD error, both having the bootstrapping structures. However, Eq. 2 has an extra challenge due to the missing reward, leading to a trivial solution of constant representation, known as complete representational collapse (Jing et al., 2021). Moreover, ZP requires distribution matching, while the Bellman equations that TD learning aims to optimize only require matching expectations.

Proposition 3 (The ℓ_2 objective Eq. 2 with *stop gradients* ($J_\ell(\phi, \theta, \overline{\phi}; h, a)$) ensures stationary points that satisfy EZP, but the ℓ_2 objective with *online* targets lacks this guarantee.)**.**

[7]Despite being a special case of probabilistic encoders, deterministic encoders deserve distinct discussion because they can be optimal in POMDPs and have been frequently used in prior works. In addition, it should be noted that a probabilistic one may help as it smooths the objective.

[8]Cosine distance is an ℓ_2 distance on the normalized vector space $\mathcal{Z} = \{ z \in \mathbb{R}^d \mid \|z\|_2 = 1\}$.

[9]The EZP condition is equivalent to the ZP condition in deterministic tasks.

Algorithm 1 Minimalist ϕ_L: learning self-predictive representations in RL

Require: Encoder $f_\phi : \mathcal{H}_t \to \mathcal{Z}$, Actor $\pi_\nu : \mathcal{Z} \to \mathcal{A}$, Critic $Q_\omega : \mathcal{Z} \times \mathcal{A} \to \mathbb{R}$, Latent Transition Model $g_\theta : \mathcal{Z} \times \mathcal{A} \to \mathcal{Z}$. Learning Rate $\alpha > 0$ and Loss Coefficient $\lambda > 0$.

1: **procedure** UPDATE(h, a, o', r)

2: Compute any model-free RL loss $\mathcal{L}_{\text{RL}}$ (based on DDPG (Lillicrap et al., 2016) here) let $Q^{\text{tar}}(h', r) := r + \gamma Q_{\overline{\omega}}(f_{\overline{\phi}}(h'), \pi_{\overline{\nu}}(f_{\overline{\phi}}(h')))$,

$$\mathcal{L}_{\text{RL}}(\phi, \omega, \nu; h', r) = (Q_\omega(f_\phi(h), a) - Q^{\text{tar}}(h', r))^2 - Q_{\overline{\omega}}(f_{\overline{\phi}}(h), \pi_\nu(f_\phi(h))). \tag{4}$$

3: Compute the auxiliary ZP loss $\mathcal{L}_{\text{aux}}(\phi, \theta; h') = \|g_\theta(f_\phi(h), a) - f_{\overline{\phi}}(h')\|_2^2$.

4: Optimize all parameters using the sum of losses:

$$[\phi, \theta, \nu, \omega] \leftarrow [\phi, \theta, \nu, \omega] - \alpha \nabla (\mathcal{L}_{\text{RL}}(\phi, \omega, \nu; h', r) + \lambda \mathcal{L}_{\text{aux}}(\phi, \theta; h')). \tag{5}$$

Prop. 3 suggests the adoption of stop-gradient targets in Eq. 2 to preserve the stationary points of EZP in both deterministic and stochastic tasks.

Theorem 3 (**Stop-gradient provably avoids representational collapse in linear models**). *Assume a linear encoder $f_\phi(h) := \phi^\top h_{-k:} \in \mathbb{R}^d$ with parameters $\phi \in \mathbb{R}^{k(|\mathcal{O}|+|\mathcal{A}|)\times d}$, which always operates on $h_{-k:}$, a recent-k truncation of history h. Assume a linear deterministic latent transition $g_\theta(z, a) := \theta_z^\top z + \theta_a^\top a \in \mathbb{R}^d$ with parameters $\theta_z \in \mathbb{R}^{d\times d}$ and $\theta_a \in \mathbb{R}^{|\mathcal{A}|\times d}$. If we train ϕ, θ using the stop-gradient ℓ_2 objective $\mathbb{E}_{h,a}\left[J_\ell(\phi, \theta, \overline{\phi}; h, a)\right]$ without RL loss, and θ relies on ϕ by reaching the stationary point with $\nabla_\theta \mathbb{E}_{h,a}\left[J_\ell(\phi, \theta, \overline{\phi}; h, a)\right] = 0$, then the matrix multiplication $\phi^\top \phi$ will retain its initial value over continuous-time training dynamics.*

Thm. 3 extends the results of (Tang et al., 2022, Theorem 1) to action-dependent latent transition, POMDP, and EMA settings. This theorem also implies that ϕ will keep full-rank during training if the initialized ϕ is full-rank[10].

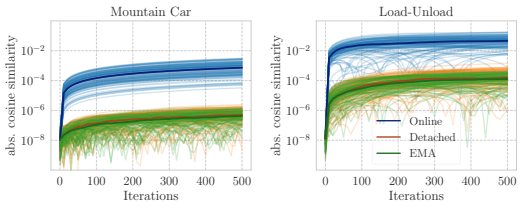

Figure 2: The absolute normalized inner product of the two column vectors in the learned encoder when using online, detached, or EMA ZP target in an MDP *(left)* and a POMDP *(right)*. We plot the results for 100 different seeds, which controls the rollouts used to sample transition and the initialization of the representation. The bold lines represent the median of the seeds.

Similar to Tang et al. (2022), we illustrate our theoretical contribution by examining the behavior of the learned encoder over time when starting from a random orthogonal initialization. We extend these results by considering both the MDP and the POMDP setting and consider two classical domains, mountain car (Moore, 1990) (MDP) and load-unload (Meuleau et al., 2013) (POMDP), where we fit an encoder ϕ with a latent state dimension of 2. Fig. 2 shows the orthogonality-preserving effect of the stop-gradient by comparing the cosine similarity between columns of the learned ϕ. As expected by Thm. 3, we see this similarity stay several orders of magnitude smaller when using stop-gradient (detached or EMA) compared to the online case. Note that although our theory discusses the continuous-time dynamics, we can approximate them with gradient steps with a small learning rate, as was done for these results.

4.3 A MINIMALIST RL ALGORITHM FOR LEARNING SELF-PREDICTIVE REPRESENTATIONS

Our theory leads to a straightforward RL algorithm that can target ϕ_L. Essentially, it integrates a single auxiliary task into any model-free RL algorithm (*e.g.*, DDPG (Lillicrap et al., 2016) and R2D2 (Kapturowski et al., 2018)), as indicated by Thm. 2. Algo. 1 provides the pseudocode for the update rule of all parameters in our algorithm given a tuple of transition data, with PyTorch code included in Appendix. The ℓ_2 ZP loss can be replaced with KL objective with probabilistic encoder and latent model, especially in stochastic environments, as suggested by Prop. 2. The $f_{\overline{\phi}}(h')$ in ZP loss stops the gradient from the encoder, following Sec. 4.2. The actor loss $-Q_{\overline{\omega}}(f_{\overline{\phi}}(h), \pi_\nu(f_\phi(h)))$ freezes the parameters of the critic and encoder, suggested by recent work on memory-based RL (Ni et al., 2023).

Our algorithm greatly simplifies prior self-predictive methods and enables a fair comparison spanning from model-free to observation-predictive representation learning. It is characterized as **minimalist** by removing reward learning, planning, multi-step predictions, projections, and metric learning. It is also **novel** by being the first to learn self-predictive representations end-to-end in POMDP literature.

Our algorithm also bridges model-free and observation-predictive representation learning. We derive learning ϕ_{Q^*} by setting the coefficient λ to 0, and learning ϕ_O by replacing ZP loss with OP loss like OFENet (Ota et al., 2020). As a by-product, by comparing $\{\phi_{Q^*}, \phi_L, \phi_O\}$ derived from

[10]This is due to the fact that $\text{rank}(A^\top A) = \text{rank}(A)$ for any real-valued matrix A.

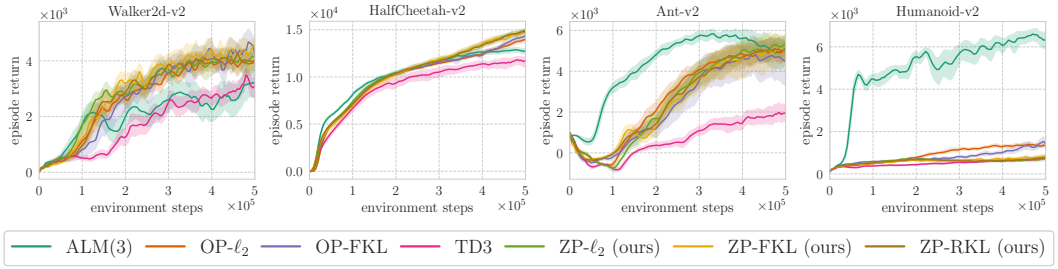

Figure 3: **Decoupling representation learning from policy optimization using our algorithm based on ALM(3) (Ghugare et al., 2022).** Comparison between ϕ_{Q^*} (TD3), ϕ_L (our algorithm (ZP-ℓ_2, ZP-FKL, ZP-RKL) and ALM(3)), ϕ_O (OP-ℓ_2, OP-FKL), in the standard MuJoCo benchmark for 500k steps, averaged over 12 seeds. The observation dimension increases from left figure to right figure $(17, 17, 111, 376)$.

our algorithm, we can **disentangle** representation learning from policy optimization because all representations are learned by the same RL algorithm. This is rarely seen in prior works, as learning ϕ_O or ϕ_L typically involves planning, while model-free RL does not. Such disentanglement allows us to examine the sample efficiency benefits derived purely from representation learning.

5 Experiments

We conduct experiments to compare RL agents learning the three representations $\{\phi_{Q^*}, \phi_L, \phi_O\}$, respectively. To decouple representation learning from policy optimization, we follow our minimalist algorithm (Algo. 1) to learn ϕ_L, and instantiate ϕ_{Q^*} and ϕ_O by setting $\lambda = 0$ and replacing ZP loss with OP loss, as we discuss in Sec. 4.3. We evaluate the algorithms in standard MDPs, distracting MDPs[11], and sparse-reward POMDPs. The experimental details are shown in Sec. E. Through the subsequent experiments, we aim to validate five hypotheses based on our theoretical insights in Sec. 3 and Sec. 4, with their motivation shown in Sec. D.1.

- **Sample efficiency hypothesis**: do the extra OP and ZP signals help ϕ_O and ϕ_L have better sample-efficiency than ϕ_{Q^*} in standard MDPs (Sec. 5.1) and *especially* in sparse-reward tasks (Sec. 5.3)?
- **Distraction hypothesis** (Sec. 5.2): since learning ϕ_O may struggle with predicting distracting observations, is it less sample-efficient than learning ϕ_L?
- **End-to-end hypothesis** (Sec. 5.3): as predicted by Thm. 2, is training an encoder end-to-end with an auxiliary task of ZP (OP) comparable to the phased training with RP and ZP (OP)?
- **ZP objective hypothesis** (Sec. 5.1, Sec. 5.2): as predicted by Prop. 1 and Prop. 2, is using ℓ_2 loss as ZP objective similar to KL loss in deterministic tasks, but not necessarily stochastic tasks?
- **ZP stop-gradient hypothesis** (Sec. 5.1, Sec. 5.3): as predicted by Thm. 3, does stop-gradient on ZP targets mitigate representational collapse compared to online ZP targets?

5.1 State Representation Learning in Standard MDPs

We first evaluate our algorithm in the relatively low-dimensional MuJoCo benchmark (Todorov et al., 2012). We implement it by simplifying ALM(3) (Ghugare et al., 2022), a state-of-the-art algorithm on MuJoCo. ALM(3) aims to learn ϕ_L end-to-end with EMA ZP target and reverse KL, shown by Thm. 2. ALM(3) requires a reward model in the encoder objective and optimizes the actor via SVG (Heess et al., 2015) with planning for 3 steps. Following our Algo. 1, we remove the reward model and multi-step predictions; instead, we train the encoder using our loss (Eq. 2 or Eq. 3) and the actor-critic is conditioned on representations and trained using model-free TD3 (Fujimoto et al., 2018), while keeping the other hyperparameters the same. These simplifications result in a 50% faster speed than ALM(3). Please see Sec. E.2 for a detailed comparison between our algorithm and ALM(3).

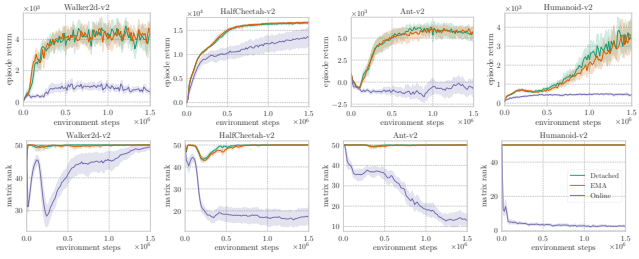

Figure 4: **Representation collapse with online targets.** On four benchmark tasks, we observe that using the online ZP target in ℓ_2 objectives results in lower returns *(top)* and low-rank representations *(bottom)*. In line with our theory, using a detached or EMA ZP target mitigates the representational collapse and yields higher returns.

Validation of sample efficiency and ZP objective hypotheses. Fig. 3 shows that our minimalist ϕ_L using EMA ZP targets can attain similar (in Ant) or *even better* (in HalfCheetah and Walker2d)

[11] Distracting MDPs refers to MDPs with distracting observations irrelevant to optimal control in this work.

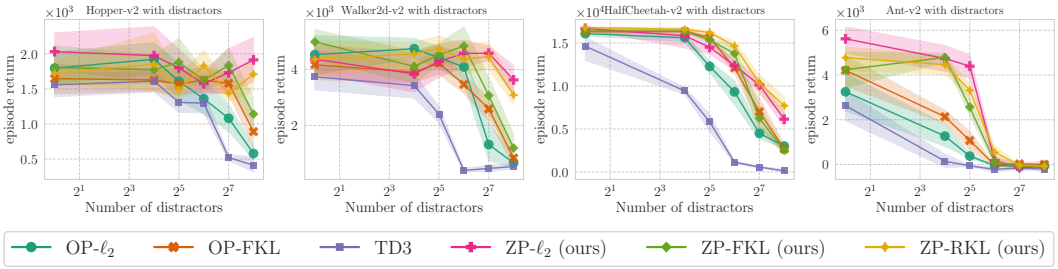

Figure 5: **Self-predictive representations are more robust.** Comparison between ϕ_{Q^*} (TD3), ϕ_L (ZP-ℓ_2, ZP-FKL, ZP-RKL) using our algorithm, ϕ_O (OP-ℓ_2, OP-FKL) in the **distracting** MuJoCo benchmark, varying the distractor dimension from 2^4 to 2^8, averaged over 12 seeds. The y-axis is final performance at 1.5M steps.

sample efficiency at 500k steps compared to ALM(3), and greatly outperforms ϕ_{Q^*}, across the entire benchmark except for Humanoid task. This suggests that the primary advantage ALM(3) brings to model-free RL in these MuJoCo tasks, lies in state representation rather than policy optimization. This benefit could be further enhanced by streamlining the algorithmic design. In Humanoid task, ALM(3)'s superior performance is likely due to SVG policy optimization's use of first-order gradient information from latent dynamics, which is particularly beneficial in high-dimensional tasks like this. In line with Prop. 1 and Prop. 2, different ZP objectives (ℓ_2, FKL, RKL) perform similarly on these tasks, which are nearly deterministic. Thus, these results support our sample efficiency and ZP objective hypotheses.

Validation of ZP stop-gradient hypothesis. We observe a significant performance degradation when switching from the stop-gradient ZP targets to *online* ones in all MuJoCo tasks for all ZP objectives (ℓ_2, FKL, RKL). Fig. 4 (top) shows the results for the ℓ_2 objective Eq. 2. To estimate the rank of the associated linearized operator for the MLP encoder, we compute the matrix rank of latent states given a batch of inputs[12], where the batch size is 512 and the latent state dimension is 50. The MLP encoders are orthogonally initialized (Saxe et al., 2013) with a full rank of 50. Fig. 4 (bottom) shows that the estimated rank of ℓ_2 objective with online targets collapses from full rank to low rank, a phenomenon known as *dimensional collapse* in self-supervised learning (Jing et al., 2021). Interestingly, the high dimensionality of a task worsens the rank collapse by comparing from left to right figures. In contrast, stop-gradient targets suffer less from rank collapse, in support to our ZP stop-gradient hypothesis. For KL objectives, switching to online targets also decreases the rank, though less severely (see Fig. 11).

5.2 State Representation Learning in Distracting MDPs

We then evaluate the robustness of representation learning by augmenting states with distractor dimensions in the MuJoCo benchmark. The distractors are i.i.d. standard isotropic Gaussians, varying the number of dimensions from 2^4 to 2^8, following the practice in Nikishin et al. (2022). The distracting task has the same optimal return as the original one and can be challenging to RL algorithms, even if model-based RL could perfectly model Gaussians. We use the same code in standard MDPs. Fig. 5 shows the final averaged returns of each algorithm (variant) in the distracting MuJoCo benchmark.

Validation of distraction hypothesis. By comparing ZP with OP objectives, with higher-dimensional distractors, learning ϕ_O degrades much faster than learning ϕ_L, verifying our distraction hypothesis. Surprisingly, model-free RL (ϕ_{Q^*}) performs worse than ϕ_O, as ϕ_{Q^*} does not need to predict the distractors. However, we still observe a severe degradation in Ant for all methods when there are 128 distractors, which we will study in the future work.

Extending ZP objective hypothesis to stochastic tasks. In stochastic tasks, Prop. 1 and Prop. 2 tell us about the (strict) upper bounds for learning EZP and ZP conditions with the practical ℓ_2 and KL objectives, respectively. Yet, they do not indicate which objective is better for stochastic tasks based on these bounds. In fact, we observe that both ℓ_2 and reverse KL perform better than forward KL in the distracting tasks, possibly because the entropy term in reverse KL smooths the training objective, and ℓ_2 objective simplifies the learning.

5.3 History Representation Learning in Sparse-Reward POMDPs

Finally, we perform an extensive empirical study on 20 MiniGrid tasks (Chevalier-Boisvert et al., 2018). These tasks, featuring partial observability and sparse rewards, serve as a rigorous test-bed for *history* representation learning. The rewards are only non-zero upon successful task completion. Episode returns are between 0 and 1, with higher returns indicating faster completion. Each task has an observation space of $7 \times 7 \times 3 = 147$ dimensions and a discrete action space of 7 options. Our

[12]The rank of an $m \times n$ matrix A is the dimension of the image of mapping $f : \mathbb{R}^n \to \mathbb{R}^m$, where $f(x) = Ax$.

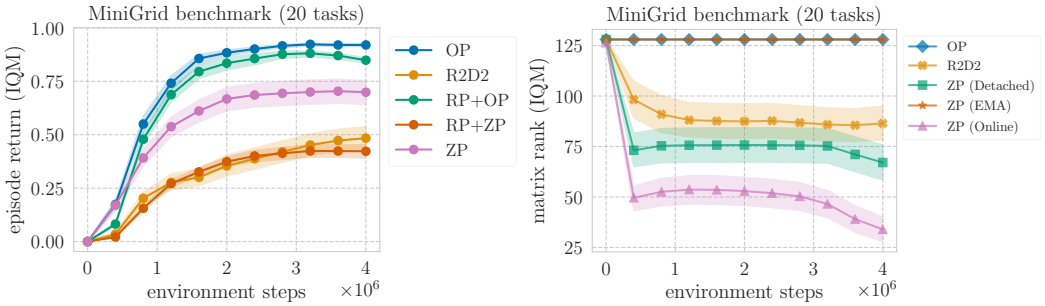

Figure 6: **End-to-end learned self-predictive and observation-predictive representations stand out in sparse-reward tasks.** We show the interquartile mean (IQM) (Agarwal et al., 2021) across **20 MiniGrid tasks**, computed over 9 seeds per task with 95% stratified bootstrap confidence intervals. The individual task plots are shown in Appendix Fig. 14 and Fig. 15. **Left**: comparison of episode returns between ϕ_{Q^*} (R2D2), ϕ_L (ZP, RP + ZP), ϕ_O (OP, RP + OP). RP + ZP and RP + OP methods are phased, while ZP and OP methods are end-to-end. **Right**: comparison of estimated matrix rank between ZP targets (online, detached, EMA), R2D2, and OP. The maximal achievable rank is 128.

model-free baseline is R2D2 (Kapturowski et al., 2018), composed of double Q-learning (Van Hasselt et al., 2016) with LSTMs (Hochreiter & Schmidhuber, 1997) as history encoders. We implement it based on recent work (Seyedsalehi et al., 2023). R2D2 uses stored LSTM hidden states for initialization, a 50-step burn-in period, and a 10-step history rollout during training. This result in a high dimensionality of $(50 + 10) \times (147 + 7) = 9240$ on histories.

Based on R2D2, we implement our algorithm by adding an auxiliary task of ZP (or OP) under ℓ_2 objectives, which is sufficient for solving these deterministic tasks (see Prop. 1). To compare the phased training (Seyedsalehi et al., 2023), we further implement methods (RP + OP, RP + ZP) that also predict rewards and freeze the encoders during Q-learning. Due to the space limit, we show the aggregated plots in Fig. 6 and defer the individual task plots to Sec. G.

Validation of sample efficiency hypothesis. By examining the aggregated learning curves of end-to-end methods in Fig. 6 left, we find that minimalist ϕ_L (ZP) significantly outperforms ϕ_{Q^*} (R2D2) on average. It, however, fall shorts of ϕ_O (OP), which aligns with our expectations given that MiniGrid tasks are deterministic with medium-dimensional clean observations. The enhanced performance of ϕ_L and ϕ_O over ϕ_{Q^*} provides empirical validation of our hypothesis in sparse-reward tasks.

Validation of end-to-end hypothesis. By comparing the aggregated learning curves between end-to-end methods and phased methods (*i.e.*, OP vs RP + OP, and ZP vs RP + ZP) in Fig. 6 left, we observe that end-to-end training (OP or ZP) yields equal or superior sample efficiency relative to phased training (RP + OP or RP + ZP) when learning observation-predictive or self-predictive representations. This supports our hypothesis, and is particularly noticeable when end-to-end learning ϕ_L (ZP) markedly excels over its phased learning counterpart (RP + ZP).

Validation of ZP stop-gradient hypothesis. We extend our rank analysis from MuJoCo to MiniGrid using the same estimation metric. As depicted in Fig. 6 right, our finding averaged across the benchmark indicates that EMA ZP targets are able to preserve their rank, while both detached and online ZP targets degrade the rank during training. Notably, while our theory does not distinguish detached and EMA targets, the observed lower rank of online targets relative to both detached and EMA targets is in line with our hypothesis. Finally, without any auxiliary task such as ZP and OP, R2D2 degrades the rank, which is predictable in the sparse-reward setting (Lyle et al., 2021).

6 Discussion

Recommendations. Based on our theoretical and empirical results, we suggest the following preliminary guidance to RL practitioners:

1. **Analyze your task first.** For example, in noisy or distracting tasks, consider using self-predictive representations. In sparse-reward tasks, consider using observation-predictive representations. In deterministic tasks, choose the deterministic ℓ_2 objectives for representation learning.
2. **Use our minimalist algorithm as your baseline.** Our algorithm allows for an independent evaluation of representation learning and policy optimization effects. Start with end-to-end learning and model-free RL for policy optimization.
3. **Implementation tips.** For our minimalist algorithm, we recommend adopting the ℓ_2 objective with EMA ZP targets first. When tackling POMDPs, start with recurrent networks as the encoder.

Please refer to Sec. D.2 for our discussion on limitations and conclusion.

ACKNOWLEDGEMENTS AND DISCLOSURE OF FUNDING

We thank Pierluca D'Oro and Zhixuan Lin for their technical help. We thank Amit Sinha, David Kanaa, David Yu-Tung Hui, Dinghuai Zhang, Doina Precup, Léo Gagnon, Pablo Samuel Castro, Raj Ghugare, Shreyas Chaudhari, and Ziyan Luo for the constructive discussion. This work was enabled by the computational resources provided by the Calcul Québec (www.calculquebec.ca) and the Digital Research Alliance of Canada (https://alliancecan.ca/), with material support from NVIDIA Corporation. This work was funded by IBM Research and Google DeepMind.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Proc. of NeurIPS*, 2021. 9, 39

Cameron Allen, Neev Parikh, Omer Gottesman, and George Konidaris. Learning markov state abstractions for deep reinforcement learning. *Proc. of NeurIPS*, 2021. 17, 33

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 37

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*. Elsevier, 1995. 5, 26

André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. *Proc. of NeurIPS*, 2017. 4, 28, 29

Torsten Bohlin. Information pattern for linear discrete-time models with stochastic coefficients. *IEEE Transactions on Automatic Control*, 1970. 1

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016. 36

Anthony R. Cassandra, Leslie Pack Kaelbling, and Michael L. Littman. Acting optimally in partially observable stochastic domains. In *Proceedings of the 12th National Conference on Artificial Intelligence, Seattle, WA, USA, July 31 - August 4, 1994, Volume 2*, 1994. 2

Pablo Samuel Castro, Prakash Panangaden, and Doina Precup. Equivalence relations in fully and partially observable markov decision processes. In *Proc. of IJCAI*, 2009. 1, 18, 19, 23, 31

Pablo Samuel Castro, Tyler Kastner, Prakash Panangaden, and Mark Rowland. Mico: Improved representations via sampling-based state similarity for markov decision processes. *Proc. of NeurIPS*, 2021. 2, 4, 33

Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic gridworld environment for openai gym. https://github.com/maximecb/gym-minigrid, 2018. 8, 37

Maxime Chevalier-Boisvert, Bolun Dai, Mark Towers, Rodrigo de Lazcano, Lucas Willems, Salem Lahlou, Suman Pal, Pablo Samuel Castro, and Jordan Terry. Minigrid & miniworld: Modular & customizable reinforcement learning environments for goal-oriented tasks. *CoRR*, 2023. 37

Era Choshen and Aviv Tamar. Contrabar: Contrastive bayes-adaptive deep rl. *arXiv preprint arXiv:2306.02418*, 2023. 33

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Proc. of NeurIPS*, 2018. 4

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015. 37

Peter Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural computation*, 1993. 1, 29

Thomas Dean and Robert Givan. Model minimization in markov decision processes. In *Proc. of AAAI*, 1997. 1

Fei Deng, Ingook Jang, and Sungjin Ahn. Dreamerpro: Reconstruction-free model-based reinforcement learning with prototypical representations. In *Proc. of ICML*. PMLR, 2022. 34

Jeffrey L Elman. Finding structure in time. *Cognitive science*, 1990. 3

Benjamin Eysenbach, Russ R Salakhutdinov, and Sergey Levine. Robust predictable control. *Proc. of NeurIPS*, 2021. 2, 4, 29

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. In *UAI*, 2004. 28

Vincent François-Lavet, Yoshua Bengio, Doina Precup, and Joelle Pineau. Combined reinforcement learning via abstract representations. In *Proc. of AAAI*, 2019. 2, 4, 27

Xiang Fu, Ge Yang, Pulkit Agrawal, and Tommi Jaakkola. Learning task informed abstractions. In *Proc. of ICML*. PMLR, 2021. 34

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *Proc. of ICML*, 2018. 7

Scott Fujimoto, Wei-Di Chang, Edward J Smith, Shixiang Shane Gu, Doina Precup, and David Meger. For sale: State-action representation learning for deep reinforcement learning. *arXiv preprint arXiv:2306.02451*, 2023. 33

Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G Bellemare. Deepmdp: Learning continuous latent space models for representation learning. In *Proc. of ICML*. PMLR, 2019. 2, 4, 5, 28

Raj Ghugare, Homanga Bharadhwaj, Benjamin Eysenbach, Sergey Levine, and Ruslan Salakhutdinov. Simplifying model-based rl: Learning representations, latent-space models, and policies with one objective. *arXiv preprint arXiv:2209.08466*, 2022. 2, 4, 5, 7, 30, 34, 36, 37

Robert Givan, Thomas Dean, and Matthew Greig. Equivalence notions and model minimization in markov decision processes. *Artificial Intelligence*, 2003. 3, 18

Zhaohan Daniel Guo, Bernardo Ávila Pires, Bilal Piot, Jean-Bastien Grill, Florent Altché, Rémi Munos, and Mohammad Gheshlaghi Azar. Bootstrap latent-predictive representations for multitask reinforcement learning. In *Proc. of ICML*, 2020. 2, 4, 28

Zhaohan Daniel Guo, Shantanu Thakoor, Miruna Pîslar, Bernardo Avila Pires, Florent Altché, Corentin Tallec, Alaa Saade, Daniele Calandriello, Jean-Bastien Grill, Yunhao Tang, et al. Byol-explore: Exploration by bootstrapped prediction. *arXiv preprint arXiv:2206.08332*, 2022. 28

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Proc. of NeurIPS*, 2018. 32

Danijar Hafner, Timothy P. Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proc. of ICML*, 2019. 4, 5, 31

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020a. 2

Danijar Hafner, Timothy P. Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *Proc. of ICLR*, 2020b. 31

Dongqi Han, Kenji Doya, and Jun Tani. Variational recurrent models for solving partially observable control tasks. In *Proc. of ICLR*, 2020. 4, 31

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. *arXiv preprint arXiv:2203.04955*, 2022. 2, 4, 5, 30, 31

Matthew J. Hausknecht and Peter Stone. Deep recurrent Q-learning for partially observable mdps. In *Proc. of AAAI*, 2015. 4

Nicolas Heess, Gregory Wayne, David Silver, Timothy Lillicrap, Tom Erez, and Yuval Tassa. Learning continuous control policies by stochastic value gradients. *Proc. of NeurIPS*, 2015. 7, 30, 37

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997. 3, 9

Masoumeh T. Izadi and Doina Precup. Using rewards for belief state updates in partially observable Markov decision processes. In *Proc. of ECML*, 2005. 2

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016. 32

Michael R James, Satinder Singh, and Michael L Littman. Planning with predictive state representations. In *Proc. of ICML*. IEEE, 2004. 31

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Proc. of NeurIPS*, 2019. 4

Nan Jiang. Notes on state abstractions. http://nanjiang.web.engr.illinois.edu/files/cs598/note4.pdf, 2018. 19

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. *arXiv preprint arXiv:2110.09348*, 2021. 5, 8

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artif. Intell.*, 1998. 3, 4, 31

Lukasz Kaiser, Mohammad Babaeizadeh, Piotr Milos, Blazej Osinski, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*, 2019. 4

Steven Kapturowski, Georg Ostrovski, John Quan, Remi Munos, and Will Dabney. Recurrent experience replay in distributed reinforcement learning. In *Proc. of ICLR*, 2018. 4, 6, 9, 38

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. of ICLR*, 2015. 37

George Konidaris, Sarah Osentoski, and Philip Thomas. Value function approximation in reinforcement learning using the fourier basis. In *Proc. of AAAI*, 2011. 1

Tejas D Kulkarni, Ardavan Saeedi, Simanta Gautam, and Samuel J Gershman. Deep successor reinforcement learning. *arXiv preprint arXiv:1606.02396*, 2016. 29

P. R. Kumar and Pravin Varaiya. *Stochastic Systems: Estimation, Identification and Adaptive Control*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1986. 1

H. Kwakernaak. *Theory of Self-Adaptive Control Systems*. Springer, 1965. 1

Moritz Lange, Noah Krystiniak, Raphael Engelhardt, Wolfgang Konen, and Laurenz Wiskott. Comparing auxiliary tasks for learning representations for reinforcement learning, 2023. URL https://openreview.net/forum?id=7Kf5_7-b7q. 32

Sascha Lange and Martin Riedmiller. Deep auto-encoder neural networks in reinforcement learning. In *Proc. of IJCNN*. IEEE, 2010. 1

Michael Laskin, Aravind Srinivas, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *Proc. of ICML*. PMLR, 2020. 33

Alex X. Lee, Anusha Nagabandi, Pieter Abbeel, and Sergey Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. In *Proc. of NeurIPS*, 2020. 4, 31

Lucas Lehnert and Michael L Littman. Successor features combine elements of model-free and model-based reinforcement learning. *The Journal of Machine Learning Research*, 2020. 4, 28, 29

Timothée Lesort, Natalia Díaz-Rodríguez, Jean-Franois Goudou, and David Filliat. State representation learning for control: An overview. *Neural Networks*, 2018. 27

Lihong Li, Thomas J Walsh, and Michael L Littman. Towards a unified theory of state abstraction for mdps. In *AIandM*, 2006. 1, 2, 3, 17, 18

Xiang Li, Jinghuan Shang, Srijan Das, and Michael Ryoo. Does self-supervised learning really improve reinforcement learning from pixels? *Proc. of NeurIPS*, 2022. 34

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 *Proc. of ICLR*, 2016. 6

Michael L. Littman, Richard S. Sutton, and Satinder P. Singh. Predictive representations of state. In *Proc. of NeurIPS*, 2001. 1, 4, 31

Clare Lyle, Mark Rowland, Georg Ostrovski, and Will Dabney. On the effect of auxiliary tasks on representation dynamics. In *Proc. of AISTATS*. PMLR, 2021. 9

Bogdan Mazoure, Remi Tachet des Combes, Thang Long Doan, Philip Bachman, and R Devon Hjelm. Deep reinforcement and infomax learning. *Proc. of NeurIPS*, 2020. 33

Nicolas Meuleau, Kee-Eung Kim, Leslie Pack Kaelbling, and Anthony R Cassandra. Solving pomdps by searching the space of finite policies. *arXiv preprint arXiv:1301.6720*, 2013. 6, 36

Aditya Mohan, Amy Zhang, and Marius Lindauer. Structure in reinforcement learning: A survey and open problems. *arXiv preprint arXiv:2306.16021*, 2023. 34

Andrew William Moore. Efficient memory-based learning for robot control. Technical report, 1990. 6, 35

Steven Morad, Ryan Kortvelesy, Matteo Bettini, Stephan Liwicki, and Amanda Prorok. Popgym: Benchmarking partially observable reinforcement learning. *arXiv preprint arXiv:2303.01859*, 2023. 1

Jelle Munk, Jens Kober, and Robert Babuška. Learning state representation for deep actor-critic control. In *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE, 2016. 1

Tianwei Ni, Benjamin Eysenbach, and Ruslan Salakhutdinov. Recurrent model-free rl can be a strong baseline for many pomdps. In *Proc. of ICML*. PMLR, 2022. 4, 33, 34

Tianwei Ni, Michel Ma, Benjamin Eysenbach, and Pierre-Luc Bacon. When do transformers shine in rl? decoupling memory from credit assignment. *arXiv preprint arXiv:2307.03864*, 2023. 6

Evgenii Nikishin, Romina Abachi, Rishabh Agarwal, and Pierre-Luc Bacon. Control-oriented model-based reinforcement learning with implicit differentiation. In *Proc. of AAAI*, 2022. 8, 36

Junhyuk Oh, Satinder Singh, and Honglak Lee. Value prediction network. *Proc. of NeurIPS*, 2017. 4, 32

Masashi Okada and Tadahiro Taniguchi. Dreaming: Model-based reinforcement learning by latent imagination without reconstruction. In *Proc. of ICRA*. IEEE, 2021. 34

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 33

Kei Ota, Tomoaki Oiki, Devesh Jha, Toshisada Mariyama, and Daniel Nikovski. Can increasing input dimensionality improve deep reinforcement learning? In *Proc. of ICML*. PMLR, 2020. 4, 6, 32

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Proc. of NeurIPS*, 2019. 38, 40

Gandharv Patil, Aditya Mahajan, and Doina Precup. On learning history based policies for controlling markov decision processes. *arXiv preprint arXiv:2211.03011*, 2022. 30

Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988. 20

Ben Poole, Sherjil Ozair, Aaron Van Den Oord, Alex Alemi, and George Tucker. On variational bounds of mutual information. In *Proc. of ICML*. PMLR, 2019. 33

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley and Sons, 1994. 16

Kate Rakelly, Abhishek Gupta, Carlos Florensa, and Sergey Levine. Which mutual-information representation learning objectives are sufficient for control? *Proc. of NeurIPS*, 2021. 33

Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013. 8

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 2020. 4, 29, 32

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. 2, 4, 5, 28, 29, 34, 35

Dino Sejdinovic, Bharath Sriperumbudur, Arthur Gretton, and Kenji Fukumizu. Equivalence of distance-based and rkhs-based statistics in hypothesis testing. *The annals of statistics*, 2013. 30

Erfan Seyedsalehi, Nima Akbarzadeh, Amit Sinha, and Aditya Mahajan. Approximate information state based convergence analysis of recurrent q-learning. *arXiv preprint arXiv:2306.05991*, 2023. 9, 30, 37, 38

Cosma Rohilla Shalizi and James P Crutchfield. Computational mechanics: Pattern and prediction, structure and simplicity. *Journal of statistical physics*, 2001. 31

Evan Shelhamer, Parsa Mahmoudieh, Max Argus, and Trevor Darrell. Loss is its own reward: Self-supervision for reinforcement learning. *arXiv preprint arXiv:1612.07307*, 2016. 32

David Silver, Hado van Hasselt, Matteo Hessel, Tom Schaul, Arthur Guez, Tim Harley, Gabriel Dulac-Arnold, David P. Reichert, Neil C. Rabinowitz, André Barreto, and Thomas Degris. The predictron: End-to-end learning and planning. In *Proc. of ICML*, 2017. 4

Satinder P Singh, Michael L Littman, Nicholas K Jong, David Pardoe, and Peter Stone. Learning predictive state representations. In *Proc. of ICML*, 2003. 1

Austin Stone, Oscar Ramirez, Kurt Konolige, and Rico Jonschkowski. The distracting control suite – a challenging benchmark for reinforcement learning from pixels. *arXiv preprint arXiv:2101.02722*, 2021. 1

Charlotte Striebel. Sufficient statistics in the optimum control of stochastic systems. *Journal of Mathematical Analysis and Applications*, 1965. 1

Jayakumar Subramanian, Amit Sinha, Raihan Seraj, and Aditya Mahajan. Approximate information state for approximate planning and reinforcement learning in partially observed systems. *J. Mach. Learn. Res.*, 2022. 1, 2, 3, 4, 16, 18, 20, 23, 24, 25, 30, 34

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 1988. 5

Richard S Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine learning proceedings 1990*. Elsevier, 1990. 4

Richard S Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. *Proc. of NeurIPS*, 1995. 1

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018. 35

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Proc. of NeurIPS*, 1999. 17

Richard S Sutton, Csaba Szepesvári, Alborz Geramifard, and Michael P Bowling. Dyna-style planning with linear function approximation and prioritized sweeping. *arXiv preprint arXiv:1206.3285*, 2012. 4

Aviv Tamar, Yi Wu, Garrett Thomas, Sergey Levine, and Pieter Abbeel. Value iteration networks. *Proc. of NeurIPS*, 2016. 4

Yunhao Tang, Zhaohan Daniel Guo, Pierre Harvey Richemond, Bernardo Ávila Pires, Yash Chandak, Rémi Munos, Mark Rowland, Mohammad Gheshlaghi Azar, Charline Le Lan, Clare Lyle, et al. Understanding self-predictive learning for reinforcement learning. *arXiv preprint arXiv:2212.03319*, 2022. 2, 6, 27

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *Proc. of IROS*, 2012. 7

Manan Tomar, Utkarsh A Mishra, Amy Zhang, and Matthew E Taylor. Learning representations for pixel-based control: What matters and why? *arXiv preprint arXiv:2111.07775*, 2021. 1, 2, 4, 5

Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proc. of AAAI*, 2016. 9

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Proc. of NeurIPS*, 2017. 3

Tingwu Wang, Xuchan Bao, Ignasi Clavera, Jerrick Hoang, Yeming Wen, Eric Langlois, Shunshi Zhang, Guodong Zhang, Pieter Abbeel, and Jimmy Ba. Benchmarking model-based reinforcement learning. *arXiv preprint arXiv:1907.02057*, 2019. 1, 34

Tongzhou Wang, Simon S Du, Antonio Torralba, Phillip Isola, Amy Zhang, and Yuandong Tian. Denoised mdps: Learning world models better than the world itself. *arXiv preprint arXiv:2206.15477*, 2022. 33

Manuel Watter, Jost Tobias Springenberg, Joschka Boedecker, and Martin A. Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In *Proc. of NeurIPS*, 2015. 1, 32

Greg Wayne, Chia-Chun Hung, David Amos, Mehdi Mirza, Arun Ahuja, Agnieszka Grabska-Barwinska, Jack Rae, Piotr Mirowski, Joel Z Leibo, Adam Santoro, et al. Unsupervised predictive memory in a goal-directed agent. *arXiv preprint arXiv:1803.10760*, 2018. 4

Lujie Yang, Kaiqing Zhang, Alexandre Amice, Yunzhu Li, and Russ Tedrake. Discrete approximate information states in partially observable environments. In *2022 American Control Conference (ACC)*. IEEE, 2022. 30

Denis Yarats, Amy Zhang, Ilya Kostrikov, Brandon Amos, Joelle Pineau, and Rob Fergus. Improving sample efficiency in model-free reinforcement learning from images. In *Proc. of AAAI*, 2021. 3, 32

Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. Mastering atari games with limited data. *Proc. of NeurIPS*, 2021. 2, 4, 5, 29

Amy Zhang, Zachary C Lipton, Luis Pineda, Kamyar Azizzadenesheli, Anima Anandkumar, Laurent Itti, Joelle Pineau, and Tommaso Furlanello. Learning causal state representations of partially observable environments. *arXiv preprint arXiv:1906.10437*, 2019. 4, 31

Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*, 2020. 2, 4, 5, 28, 34

Jin Zhang, Jianhao Wang, Hao Hu, Tong Chen, Yingfeng Chen, Changjie Fan, and Chongjie Zhang. Metacure: Meta reinforcement learning with empowerment-driven exploration. In *Proc. of ICML*, 2021. 34

Yi Zhao, Wenshuai Zhao, Rinu Boney, Juho Kannala, and Joni Pajarinen. Simplified temporal consistency reinforcement learning. *arXiv preprint arXiv:2306.09466*, 2023. 4, 31, 34

Luisa M. Zintgraf, Leo Feng, Cong Lu, Maximilian Igl, Kristian Hartikainen, Katja Hofmann, and Shimon Whiteson. Exploration in approximate hyper-state space for meta reinforcement learning. In *Proc. of ICML*, 2021. 34

Appendix

Table of Contents

A A Unified View on State and History Representations

A.1 Notation

Table 2 shows the glossary used in this paper.

A.2 Additional Background

Remark on the latent state distribution. In this paper, we assume the latent space $\mathcal{Z}$ as a pre-specified Banach space, which is a complete normed vector space. We further assume any latent state distribution defined on $\mathcal{Z}$ has a finite expectation. To avoid a measure-theoretic treatment, we assume that Z is discrete-valued in our proof analysis. The proof arguments are easily generalized to the case when $\mathcal{Z}$ lies in a Banach space using standard arguments.

Remark on the existence of optimal value and policy in POMDPs. In an MDP, it is well-known that there exists a unique optimal value function following the Bellman equation, which induces an optimal deterministic policy (Puterman, 1994). In POMDPs, the result is complicated. For a *finite-horizon* POMDP, one can construct a finite-dimensional state space by stacking all previous observations and actions to convert a POMDP into an MDP, thus the MDP result can be directly applied. For an *infinite-horizon* POMDP, Subramanian et al. (2022, Theorem 25) shows that the unique optimal value function exists when the POMDP has a time-invariant finite-dimensional information

Table 2: **Glossary of notations** used in this paper.

Notation	Text description	Math description
γ	Discount factor	$\gamma \in [0, 1]$
T	Horizon	$T \in \mathbb{N} \cup \{+\infty\}$
s_t	State at step t	$s \in \mathcal{S}$
o_t	Observation at step t	$o \in \mathcal{O}$
a_t	Action at step t	$a \in \mathcal{A}$
r_t	Reward at step t	$r \in \mathbb{R}$
h_t	History at step t	$h_t = (h_{t-1}, a_{t-1}, o_t) \in \mathcal{H}_t, h_1 = o_1$
$P(o_{t+1} \mid h_t, a_t)$	Environment transition	
$R(h_t, a_t)$	Environment reward function	$R : \mathcal{H}_t \times \mathcal{A} \to \Delta(\mathbb{R})$
$\pi(a_t \mid h_t)$	Policy (actor)	
$\pi^*(h_t)$	Optimal policy (actor)	
$Q^\pi(h_t, a_t)$	Value (critic)	
$Q^*(h_t, a_t)$	Optimal value (critic)	
ϕ	Encoder of history	$\phi : \mathcal{H}_t \to \mathcal{Z}$
z_t	Latent state at step t	$z_t = \phi(h_t) \in \mathcal{Z}$
$P_z(z_{t+1} \mid z_t, a_t)$	Latent transition	
$R_z(z_t, a_t)$	Latent reward function	
$\pi_z(a_t \mid z_t)$	Latent policy (actor)	
$\pi_z^*(z_t)$	Optimal latent policy (actor)	
$Q_z^{\pi_z}(z_t, a_t)$	Latent value (critic)	
$Q_z^*(z_t, a_t)$	Optimal latent value (critic)	
RP	Expected **R**eward **P**rediction	$\mathbb{E}[r_t \mid h_t, a_t] = R_z(\phi(h_t), a_t)$
OR	**O**bservation **R**econstruction	$o_t = \psi_o(\phi(h_t))$
OP	Next **O**bservation **P**rediction	$P(o_{t+1} \mid h_t, a_t) = P_o(o_{t+1} \mid \phi(h_t), a_t)$
ZP	Next Latent State z **P**rediction	$P(z_{t+1} \mid h_t, a_t) = P_z(z_{t+1} \mid \phi(h_t), a_t)$
EZP	**E**xpected Next Latent State z **P**rediction	$\mathbb{E}[z_{t+1} \mid h_t, a_t] = \mathbb{E}[z_{t+1} \mid \phi(h_t), a_t]$
Rec	Recurrent Encoder	$\phi(h_{t+1}) = \psi_z(\phi(h_t), a_t, o_{t+1})$
ZM	Markovian Latent Transition	$z_{t+1} \perp\!\!\!\perp z_{1:t-1}, a_{1:t-1} \mid \phi(h_t), a_t$
ϕ_{π^*}	π^*-irrelevance abstraction	$\phi(h_1) = \phi(h_2) \implies \pi^*(h_1) = \pi^*(h_2)$
ϕ_{Q^*}	Q^*-irrelevance abstraction	$\phi(h_1) = \phi(h_2) \implies Q^*(h_1, a) = Q^*(h_2, a)$
ϕ_M	Markovian abstraction	RP + ZM
ϕ_L	Self-predictive abstraction	RP + ZP $\iff$ ϕ_{Q^*} + ZP
ϕ_O	Observation-predictive abstraction	RP + OP + Rec $\iff$ ϕ_{Q^*} + OP + Rec

state, which is the case when the unobserved state space is finite. The POMDP experiments shown in Sec. 5.3 satisfy this assumption because they have a finite state space. For POMDPs with infinite-dimensional information states, the result remains unclear.

A.2.1 Additional Abstractions and Formalizing the Relationship

First, we present two additional abstractions not shown in the main paper, which are also used in prior work. Then we formalize Thm. 1 with Thm. 4 using the concept of granularity in relation.

π^*-irrelevance abstraction. An encoder ϕ_{π^*} yields a π^*-irrelevance abstraction (Li et al., 2006) if it contains the necessary information (a "sufficient statistics") for selecting return-maximizing actions. Formally, if $\phi_{\pi^*}(h_i) = \phi_{\pi^*}(h_j)$ for some $h_i, h_j \in \mathcal{H}_t$, then $\pi^*(h_i) = \pi^*(h_j)$. One way of obtaining a π^*-irrelevance abstraction is to learn an encoder ϕ end-to-end with a policy $\pi_z(a \mid \phi(h))$ by model-free RL (Sutton et al., 1999) such that $\pi_z^*(\phi_{\pi^*}(h)) = \pi^*(h), \forall h$.

Markovian abstraction. An encoder ϕ_M provides Markovian abstraction if it satisfies the expected reward condition RP and **Markovian latent transition (ZM)** condition: for any $z_k = \phi_M(h_k)$,

$$P(z_{t+1} \mid z_{1:t}, a_{1:t}) = P(z_{t+1} \mid z_t, a_t), \quad \forall z_{1:t+1}, a_{1:t}. \tag{ZM}$$

This extends Markovian abstraction (Allen et al., 2021) in MDPs to POMDPs.

Granularity in relation. In MDPs, it is well-known that state representations form a hierarchical structure (Li et al., 2006, Theorem 2), but this idea had not been extended to the POMDP case. We do so here by defining an equivalent concept of "granularity". We say that an encoder ϕ_A is finer than or equal to another encoder ϕ_B, denoted as $\phi_A \succeq \phi_B$, if and only if for any histories $h_i, h_j \in \mathcal{H}_t$, $\phi_A(h_i) = \phi_A(h_j)$ implies $\phi_B(h_i) = \phi_B(h_j)$. The relation $\succeq$ is a partial ordering. Using this notion, we can show Thm. 4:

Theorem 4 (Granularity of state and history abstractions (the formal version of Thm. 1)). $\phi_O \succeq \phi_L \succeq \phi_{Q^*} \succeq \phi_{\pi^*}$.

Abstract MDP. Given an encoder ϕ, we can construct an abstract MDP (Li et al., 2006) $\mathcal{M}_\phi = (\mathcal{Z}, \mathcal{A}, P_z, R_z, \gamma, T)$ for a POMDP $\mathcal{M}_O$. The latent reward R_z and latent transition P_z are then given by: $R_z(z,a) = \int P(h \mid z)\mathbb{E}[r \mid h,a]dh$, $P_z(z' \mid z,a) = \int P(h \mid z)P(o' \mid h,a)\delta(z' = \phi(h'))dhdo'$, where $P(h \mid z) = 0$ for any $\phi(h) \neq z$ and is normalized to a distribution. The optimal latent (Markovian) value function $Q_z^*(z,a)$ satisfies $Q_z^*(z,a) = R_z(z,a) + \gamma\mathbb{E}_{z' \sim P_z(|z,a)}[\max_{a'} Q_z^*(z',a')]$, and the optimal latent policy $\pi_z^*(z) = \operatorname{argmax}_a Q_z^*(z,a)$. It is important to note that this definition focuses solely on the process by which the encoder induces a corresponding abstract MDP, without addressing the quality of the encoder itself.

A.2.2 Alternative Definitions

In the main paper (Sec. 2), we present the concepts of self-predictive abstraction ϕ_L and observation-predictive abstraction ϕ_O. In most prior works, these concepts were defined in an alternative way – using a pair of states (histories). In comparison, our definition is based on a pair of a state (history) and a latent state, which we believe is more comprehensible and help derive the auxiliary objectives.

For completeness, here we restate their definition, extended to POMDPs, and then show the equivalence between their and our definitions.

Model-irrelevance abstraction (Li et al., 2006) (bisimulation relation (Givan et al., 2003)) Φ_L. If for any two histories $h_i, h_j \in \mathcal{H}$ such that $\Phi_L(h_i) = \Phi_L(h_j)$, then

$$\mathbb{E}[r \mid h_i, a] = \mathbb{E}[r \mid h_j, a], \quad \forall a \in \mathcal{A}, \tag{6}$$

$$P(z' \mid h_i, a) = P(z' \mid h_j, a), \quad \forall a \in \mathcal{A}, z' \in \mathcal{Z}, \tag{7}$$

where $P(z' \mid h, a) = \int P(o' \mid h, a)\delta(z' = \Phi_L(h'))do'$. Here we extend the concept from MDPs (Li et al., 2006; Givan et al., 2003) into POMDPs. It is worth noting that while original concepts assume deterministic rewards or require reward distribution matching for stochastic rewards (Castro et al., 2009) in Eq. 6, the requirement can indeed be relaxed. As shown by Subramanian et al. (2022), it is sufficient to ensure expected reward matching to maintain optimal value functions. As such, we adopt this relaxed requirement of expectation matching in our concept.

Proposition 4 (Φ_L is equivalent to ϕ_L).

Proof. It is easy to see that ϕ_L implies Φ_L. If $\phi_L(h_i) = \phi_L(h_j)$, then by RP,

$$\mathbb{E}[r \mid h_i, a] = R_z(\phi_L(h_i), a) = R_z(\phi_L(h_j), a) = \mathbb{E}[r \mid h_j, a], \tag{8}$$

and by ZP,

$$P(z' \mid h_i, a) = P_z(z' \mid \phi_L(h_i), a) = P_z(z' \mid \phi_L(h_j), a) = P(z' \mid h_j, a). \tag{9}$$

Therefore, ϕ_L implies Φ_L.

Now we want to show Φ_L implies ϕ_L. We will use the following fact: for any two random variables X, Y and a function f that maps Y into a random variable Z, we have $X \perp\!\!\!\perp f(Y) \mid Y$. This is equivalent to say:

$$P(X = x \mid Y = y) = P(X = x \mid Y = y, Z = f(y)), \quad \forall x, y. \tag{10}$$

A corollary is on the conditional expectation:

$$\mathbb{E}[X \mid Y = y] = \mathbb{E}[X \mid Y = y, Z = f(y)]. \tag{11}$$

First, to see RP condition: using the fact (Eq. 11),

$$\mathbb{E}[r \mid \mathcal{H} = h_i, \mathcal{A} = a] = \mathbb{E}[r \mid \mathcal{H} = h_i, \mathcal{A} = a, \mathcal{Z} = \phi(h_i)]. \tag{12}$$

By Eq. 6, we have for any h_i, h_j such that $\phi(h_i) = \phi(h_j) := z$,

$$\mathbb{E}[r \mid \mathcal{H} = h_i, \mathcal{A} = a, \mathcal{Z} = z] = \mathbb{E}[r \mid \mathcal{H} = h_j, \mathcal{A} = a, \mathcal{Z} = z]. \tag{13}$$

This exactly indicates RP condition: $\mathbb{E}[r \mid \mathcal{H} = h_i, \mathcal{A} = a]$ is a function of $\phi(h_i), a$.

Similar to the proof of showing RP, we can show ZP: using the fact (Eq. 10),

$$P(z' \mid \mathcal{H} = h_i, \mathcal{A} = a) = P(z' \mid \mathcal{H} = h_i, \mathcal{A} = a, \mathcal{Z} = \phi(h_i)). \tag{14}$$

By Eq. 7, we have for any h_i, h_j such that $\phi(h_i) = \phi(h_j) := z$,

$$P(z' \mid \mathcal{H} = h_i, \mathcal{A} = a, \mathcal{Z} = z) = P(z' \mid \mathcal{H} = h_j, \mathcal{A} = a, \mathcal{Z} = z). \tag{15}$$

This exactly indicates ZP condition: $P(z' \mid \mathcal{H} = h_i, \mathcal{A} = a)$ is a distribution conditioned on $\phi(h_i), a$.

□

Belief abstraction Φ_O (weak belief bisimulation relation (Castro et al., 2009)). It satisfies Rec, and if for any two histories $h_i, h_j \in \mathcal{H}$ such that $\Phi_O(h_i) = \Phi_O(h_j)$, then

$$\mathbb{E}[r \mid h_i, a] = \mathbb{E}[r \mid h_j, a], \quad \forall a \in \mathcal{A}, \tag{16}$$

$$P(o' \mid h_i, a) = P(o' \mid h_j, a), \quad \forall a \in \mathcal{A}, o' \in \mathcal{O}. \tag{17}$$

This concept is known as a naive abstraction in MDPs (Jiang, 2018) and weak belief bisimulation relation in POMDPs (Castro et al., 2009). Similarly, prior concepts assume deterministic reward or distribution matching for stochastic rewards, while we relax it to expected reward matching.

Proposition 5 (Φ_O is equivalent to ϕ_O).

Proof. The proof is almost the same as the proof of Prop. 4 by replacing z' with o'. □

A.3 Propositions and Proofs

With the additional background in Sec. A.2, we show the complete implication graph in Fig. 7 built on Fig. 1.

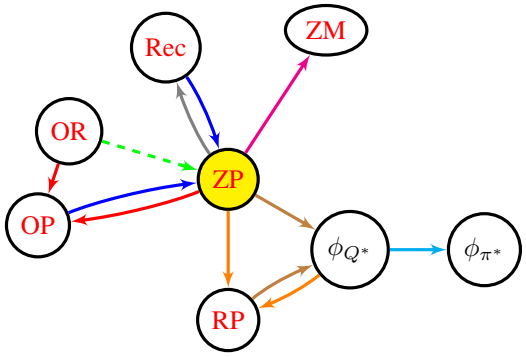

Figure 7: **The complete implication graph** showing the relations between the conditions on history representations. The source nodes of the edges with the same color together imply the target node. The dashed edge means it only applies to MDPs. As a quick reminder, RP: expected reward prediction, OP: next observation prediction, OR: observation reconstruction, ZP: next latent state prediction, Rec: recurrent encoder, ZM: Markovian latent transition. All the connections are discovered in this work, except for (1) OP + Rec implying ZP, (2) ZP + RP implying ϕ_{Q^*}, (3) ϕ_{Q^*} implying ϕ_{π^*}.

A.3.1 Results Related to ZP

Lemma 1 (Functions of independent random variables are also independent). *If $X \perp\!\!\!\perp Y$, then for any functions f, g, we have $f(X) \perp\!\!\!\perp g(Y)$.*

Proof. This is a well-known result. Here is an elementary proof. Let A, B be any two sets,

$$P(f(X) \in A, g(Y) \in B) = P(X \in f^{-1}(A), Y \in g^{-1}(B)) \tag{18}$$

$$\overset{X \perp\!\!\!\perp Y}{=} P(X \in f^{-1}(A))P(Y \in g^{-1}(B)) = P(f(X) \in A)P(g(Y) \in B). \tag{19}$$

□

Lemma 2. *If $X \perp\!\!\!\perp Y \mid Z$, then for any function f, we have $X \perp\!\!\!\perp Y, f(Z) \mid Z$.*

Proof.

$$P(Y, f(Z) \mid X, Z) = P(f(Z) \mid X, Z)P(Y \mid X, Z, f(Z)) \tag{20}$$

$$= P(f(Z) \mid Z)P(Y \mid Z) = P(f(Z) \mid Z)P(Y \mid Z, f(Z)) = P(Y, f(Z) \mid Z). \tag{21}$$

□

Proposition 6 (ZP implies both ZM and Rec.).

Remark 1. These are **new** results. ZP implying ZM means $\phi_L \succeq \phi_M$.

*Proof of **Prop. 6** (**ZP implies ZM**).* Since ZP that $z_{t+1} \perp\!\!\!\perp h_t \mid z_t, a_t$, this implies $z_{t+1} \perp\!\!\!\perp f(h_t) \mid z_t, a_t$ for any transformation f by Lemma 1. One special case of f is that $f(h_t) = (z_{1:t}, a_{1:t-1})$, where $z_k = \phi(h_k)$, which is ZM. □

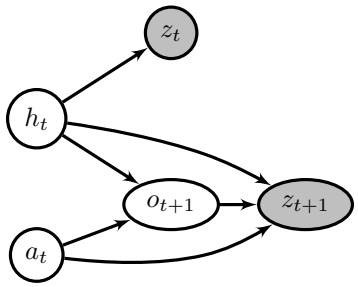

Figure 8: **The graphical model of the interaction between history encoder and the environment.**

*Proof of **Prop. 6** (**ZP implies Rec**).* Let ϕ satisfy ZP, *i.e.* $z' \perp\!\!\!\perp h \mid \phi(h), a$. Then we can show that $z' \perp\!\!\!\perp h \mid \phi(h), a, o'$. This is because the graphical model $((h, a) \to o'$ and $(h, a, o') \to z'$; see Fig. 8) does not have v-structure such that $(h, z') \to o'$, thus adding the variable o' to conditionals preserves conditional independence, by the principle of d-separation (Pearl, 1988).

As (a, o') also appears in the condition, we have $z' \perp\!\!\!\perp h' \mid \phi(h), a, o'$ by Lemma 2, which is the probabilistic form of Rec. □

Proposition 7 (OP and Rec imply ZP (Subramanian et al., 2022).).

*Proof of **Prop. 7** and **Thm. 4** ($\phi_O \succeq \phi_L$).* We directly follow the proof in (Subramanian et al., 2022, Proposition 4). Let ϕ satisfy OP and Rec, then we will have ZP:

$$P(z' \mid h, a) = \int P(z', o' \mid h, a)do' = \int \delta(z' = \phi(h'))P(o' \mid h, a)do' \tag{22}$$

$$\overset{(Rec, OP)}{=} \int \delta(z' = \psi_z(\phi(h), a, o'))P_o(o' \mid \phi(h), a)do' \tag{23}$$

$$= \int P(z', o' \mid \phi(h), a)do' = P_z(z' \mid \phi(h), a). \tag{24}$$

□

Proof of Thm. 4 ($\phi_{Q^*} \succeq \phi_{\pi^*}$). If $\phi_{Q^*}(h_i) = \phi_{Q^*}(h_j)$, then $Q^*(h_i, a) = Q^*(h_j, a), \forall a$, and then taking argmax we get the optimal policy, $\pi^*(h_1) = \operatorname{argmax}_a Q^*(h_1, a) = \operatorname{argmax}_a Q^*(h_2, a) = \pi^*(h_2)$. □

Proposition 8 (OR and ZP imply OP).

Proof. Consider given h, a, for any o',

$$P(o', \phi(h') \mid h, \phi(h), a) = P(\phi(h') \mid h, a) P(o' \mid \phi(h'), h, \phi(h), a) \tag{25}$$

$$\overset{(ZP,OR)}{=} P_z(\phi(h') \mid \phi(h), a) \delta(o' = \psi_o(\phi(h'))) \tag{26}$$

$$= P_z(\phi(h') \mid \phi(h), a) P(o' \mid \phi(h'), \phi(h), a) \tag{27}$$

$$= P(o', \phi(h') \mid \phi(h), a), \tag{28}$$

where Ln. 27 follows that the OR condition $o' \perp\!\!\!\perp h' \mid \phi(h')$ implies $o' \perp\!\!\!\perp \phi(h), a \mid \phi(h')$ by Lemma 1. Therefore, $o', \phi(h') \perp\!\!\!\perp h \mid \phi(h), a$. By Lemma 1, we have OP $o' \perp\!\!\!\perp h \mid \phi(h), a$.

□

Proposition 9 (In MDPs, OR implies ZP and OP).

Proof. Assume ϕ satifies OR in MDPs, *i.e.* there exists $\psi_s : \mathcal{Z} \to \mathcal{S}$ such that $\psi_s(\phi(s)) = s$. We want to show that $z', s \perp\!\!\!\perp s \mid \phi(s), a$ which implies ZP by Lemma 1. In fact,

$$P(z', s \mid s, \phi(s), a) = P(z', s \mid s, a) = P(z' \mid s, a) \delta(s = s) \tag{29}$$

$$P(z', s \mid \phi(s), a) = P(s \mid \phi(s), a) P(z' \mid s, \phi(s), a) \tag{30}$$

$$\overset{OR}{=} \delta(s = s) P(z' \mid s, a). \tag{31}$$

Similar proof to OP by replacing z' with s'. □

A.3.2 Results Related to Multi-Step Conditions

Below are results on multi-step RP, ZP, and OP, and due to space limit, we do not show these connections in Fig. 7.

Proposition 10 (ZP is equivalent to multi-step ZP). *For $k \in \mathbb{N}^+$, define k-step ZP as*

$$P(z_{t+k} \mid h_t, a_{t:t+k-1}) = P(z_{t+k} \mid \phi(h_t), a_{t:t+k-1}), \quad \forall h, a, z. \tag{32}$$

Proof. As ZP is 1-step ZP, thus multi-step ZP implies ZP. Now we show that ZP implies multi-step ZP.

$$P(z_{t+k} \mid h_t, a_{t:t+k-1}) = \int P(z_{t+1:t+k}, o_{t+1:t+k} \mid h_t, a_{t:t+k-1}) do_{t+1:t+k} dz_{t+1:t+k-1} \tag{33}$$

$$= \int \prod_{i=1}^{k} \delta(z_{t+i} = \phi(h_{t+i})) P(o_{t+i} \mid h_{t+i-1}, a_{t+i-1}) do_{t+1:t+k} dz_{t+1:t+k-1} \tag{34}$$

$$= \int \left(\int \delta(z_{t+k} = \phi(h_{t+k})) P(o_{t+k} \mid h_{t+k-1}, a_{t+k-1}) do_{t+k} \right) \tag{35}$$

$$\prod_{i=1}^{k-1} \delta(z_{t+i} = \phi(h_{t+i})) P(o_{t+i} \mid h_{t+i-1}, a_{t+i-1}) do_{t+1:t+k-1} dz_{t+1:t+k-1} \tag{36}$$

$$= \int P(z_{t+k} \mid h_{t+k-1}, a_{t+k-1}) \prod_{i=1}^{k-1} \delta(z_{t+i} = \phi(h_{t+i})) P(o_{t+i} \mid h_{t+i-1}, a_{t+i-1}) do_{t+1:t+k-1} dz_{t+1:t+k-1} \tag{37}$$

$$\overset{ZP}{=} \int P(z_{t+k} \mid \phi(h_{t+k-1}), a_{t+k-1}) \prod_{i=1}^{k-1} \delta(z_{t+i} = \phi(h_{t+i})) P(o_{t+i} \mid h_{t+i-1}, a_{t+i-1}) do_{t+1:t+k-1} dz_{t+1:t+k-1} \tag{38}$$

$$= \int P(z_{t+k} \mid z_{t+k-1}, a_{t+k-1}) \prod_{i=1}^{k-1} \delta(z_{t+i} = \phi(h_{t+i})) P(o_{t+i} \mid h_{t+i-1}, a_{t+i-1}) do_{t+1:t+k-1} dz_{t+1:t+k-1} \tag{39}$$

$$= \dots \tag{40}$$

$$= \int \prod_{i=2}^{k} P(z_{t+i} \mid z_{t+i-1}, a_{t+i-1}) P(z_{t+1} \mid h_t, a_t) dz_{t+1:t+k-1} \tag{41}$$

$$\overset{ZP}{=} \int \prod_{i=2}^{k} P(z_{t+i} \mid z_{t+i-1}, a_{t+i-1}) P(z_{t+1} \mid \phi(h_t), a_t) dz_{t+1:t+k-1} \tag{42}$$

$$\overset{ZM}{=} \int P(z_{t+1:t+k} \mid \phi(h_t), a_{t:t+i-1}) dz_{t+1:t+k-1} \tag{43}$$

$$= P(z_{t+k} \mid \phi(h_t), a_{t:t+i-1}). \tag{44}$$

□

Proposition 11 (ZP and RP imply multi-step RP). *For $k \in \mathbb{N}^+$, define k-step RP as*

$$\mathbb{E}[r_{t+k} \mid h_t, a_{t:t+k}] = \mathbb{E}[r_{t+k} \mid \phi(h_t), a_{t:t+k}], \quad \forall h, a \tag{45}$$

Proof.

$$\mathbb{E}[r_{t+k} \mid h_t, a_{t:t+k}] = \int P(o_{t+1:t+k} \mid h_t, a_{t:t+k-1}) \mathbb{E}[r_{t+k} \mid h_{t+k}, a_{t+k}] do_{t+1:t+k} \tag{46}$$

$$\overset{RP}{=} \int P(o_{t+1:t+k} \mid h_t, a_{t:t+k-1}) R_z(\phi(h_{t+k}), a_{t+k}) do_{t+1:t+k} \tag{47}$$

$$= \int P(o_{t+1:t+k} \mid h_t, a_{t:t+k-1}) \delta(z_{t+k} = \phi(h_{t+k})) R_z(z_{t+k}, a_{t+k}) do_{t+1:t+k} dz_{t+k} \tag{48}$$

$$= \int \left(\int P(o_{t+1:t+k} \mid h_t, a_{t:t+k-1}) \delta(z_{t+k} = \phi(h_{t+k})) do_{t+1:t+k} \right) R_z(z_{t+k}, a_{t+k}) dz_{t+k} \tag{49}$$

$$= \int P(z_{t+k} \mid h_t, a_{t:t+k-1}) R_z(z_{t+k}, a_{t+k}) dz_{t+k} \tag{50}$$

$$\overset{k\text{-step } ZP}{=} \int P(z_{t+k} \mid \phi(h_t), a_{t:t+k-1}) R_z(z_{t+k}, a_{t+k}) dz_{t+k} \tag{51}$$

$$= \mathbb{E}[r_{t+k} \mid \phi(h_t), a_{t:t+k}], \tag{52}$$

where k-step ZP is implied by ZP by Prop. 10. □

Proposition 12 (OP implies multi-step OP in MDPs, but not POMDPs). *For $k \in \mathbb{N}^+$, define k-step OP as*

$$P(o_{t+k} \mid h_t, a_{t:t+k-1}) = P(o_{t+k} \mid \phi(h_t), a_{t:t+k-1}), \quad \forall h, a, o. \tag{53}$$

Proof. We first show the result in MDPs. Assume a state encoder ϕ satisfies OP,

$$P(s_{t+k} \mid s_t, a_{t:t+k-1}) = \int P(s_{t+1:t+k} \mid s_t, a_{t:t+k-1}) ds_{t+1:t+k-1} \tag{54}$$

$$\overset{\text{MDPs}}{=} \int P(s_{t+1} \mid s_t, a_t) \prod_{i=2}^{k} P(s_{t+i} \mid s_{t+i-1}, a_{t+i-1}) ds_{t+1:t+k-1} \tag{55}$$

$$\overset{OP}{=} \int P(s_{t+1} \mid \phi(s_t), a_t) \prod_{i=2}^{k} P(s_{t+i} \mid s_{t+i-1}, a_{t+i-1}) ds_{t+1:t+k-1} \tag{56}$$

$$\overset{\text{MDPs}}{=} \int P(s_{t+1:t+k} \mid \phi(s_t), a_{t:t+k-1}) ds_{t+1:t+k-1} \tag{57}$$

$$= P(s_{t+k} \mid \phi(s_t), a_{t:t+k-1}). \tag{58}$$

However, in POMDPs, OP does not imply multi-step OP. This can be shown by a counterexample in Castro et al. (2009, Theorem 4.10), where the weak belief bisimulation relation corresponds to single-step OP and RP, while trajectory equivalence corresponds to multi-step OP and RP. The idea is to show that for two histories h_t^1 and h_t^2, if $P(o_{t+1} \mid h_t^1, a_t) = P(o_{t+1} \mid h_t^2, a_t), \forall o_{t+1}, a_t$, it does not imply that $P(o_{t+2} \mid h_t^1, a_t, o_{t+1}, a_{t+1}) = P(o_{t+2} \mid h_t^2, a_t, o_{t+1}, a_{t+1}), \forall o_{t+1:t+2}, a_{t:t+1}$. □

A.3.3 Results Related to ϕ_{Q^*}

*Proof sketch of **ZP + RP** (ϕ_L) **imply** ϕ_{Q^*}.* To show $Q^*(h,a) = Q_z^*(\phi_L(h), a), \forall h, a$, please see (Subramanian et al., 2022, Theorem 5 and Theorem 25) for finite-horizon and infinite-horizon POMDPs, respectively. For the approximate version, please see (Subramanian et al., 2022, Theorem 9 and Theorem 27). By definition, $Q^*(h,a) = Q_z^*(\phi_L(h), a), \forall h, a$ implies that ϕ_L is a kind of ϕ_{Q^*}. □

*Proof of **Thm. 2** (**ZP + ϕ_{Q^*} imply RP**).* Suppose ϕ satisfies ZP and we train model-free RL with value parameterized by $\mathcal{Q}(\phi(h), a)$ to satisfy the Bellman optimality equation:

$$\mathcal{Q}(\phi(h_t), a_t) = \begin{cases} \mathbb{E}[r_t \mid h_t, a_t] & t = T, \\ \mathbb{E}[r_t \mid h_t, a_t] + \gamma \mathbb{E}_{o_{t+1} \sim P(\mid h_t, a_t)} \left[\max_{a_{t+1}} \mathcal{Q}(\phi(h_{t+1}), a_{t+1}) \right] & \text{else}, \end{cases} \tag{59}$$

where the case $t = T$ only applies to finite-horizon problems (the same below). This is equivalent to say that $\mathcal{Q}(\phi(h_t), a_t) = Q^*(h_t, a_t), \forall h_t, a_t$, where Q^* satisfies the Bellman optimality equation, too:

$$Q^*(h_t, a_t) = \begin{cases} \mathbb{E}[r_t \mid h_t, a_t] & t = T, \\ \mathbb{E}[r_t \mid h_t, a_t] + \gamma \mathbb{E}_{o_{t+1} \sim P(\mid h_t, a_t)} \left[\max_{a_{t+1}} Q^*(h_{t+1}, a_{t+1}) \right] & \text{else}. \end{cases} \tag{60}$$

Now we can construct an abstract MDP with ϕ. The latent transition matches due to ZP. The latent reward function is purely defined by latent value and latent transition[13]:

$$\mathcal{R}_z(z_t, a_t) := \begin{cases} \mathcal{Q}(z_t, a_t) & t = T, \\ \mathcal{Q}(z_t, a_t) - \gamma \mathbb{E}_{z_{t+1} \sim P(\mid z_t, a_t)} \left[\max_{a_{t+1}} \mathcal{Q}(z_{t+1}, a_{t+1}) \right] & \text{else}. \end{cases} \tag{61}$$

We want to show RP condition: $\mathcal{R}_z(\phi(h_t), a_t) = \mathbb{E}[r_t \mid h_t, a_t], \forall h_t, a_t$.

Here is our proof. Recall that the grounded reward function can also be derived reversely by Q^*:

$$\mathbb{E}[r_t \mid h_t, a_t] := \begin{cases} Q^*(h_t, a_t) & t = T, \\ Q^*(h_t, a_t) - \gamma \mathbb{E}_{o_{t+1} \sim P(\mid h_t, a_t)} \left[\max_{a_{t+1}} Q^*(h_{t+1}, a_{t+1}) \right] & \text{else}. \end{cases} \tag{62}$$

If the problem is finite-horizon with horizon T and when $t = T$, RP holds due to $\mathcal{Q}(\phi(h_T), a_T) = Q^*(h_T, a_T)$.

Now consider general case when $t < T$ in finite-horizon ($\gamma = 1$) and any t in infinite-horizon ($\gamma < 1$). Due to Q-value match ($\mathcal{Q}(\phi(h_t), a_t) = Q^*(h_t, a_t)$), it is equivalent to show that

$$\mathbb{E}_{o_{t+1} \sim P(\mid h_t, a_t)} \left[\max_{a_{t+1}} Q^*(h_{t+1}, a_{t+1}) \right] = \mathbb{E}_{z_{t+1} \sim P(\mid \phi(h_t), a_t)} \left[\max_{a_{t+1}} \mathcal{Q}(z_{t+1}, a_{t+1}) \right], \quad \forall h_t, a_t, t. \tag{63}$$

Proof for this:

$$\text{LHS} \overset{\phi_{Q^*}}{=} \mathbb{E}_{o_{t+1} \sim P(\mid h_t, a_t)} \left[\max_{a_{t+1}} \mathcal{Q}(\phi(h_{t+1}), a_{t+1}) \right] \tag{64}$$

[13] In the main paper, we omit the finite-horizon case due to space limit.

$$= \int \left(\int P(o_{t+1} \mid h_t, a_t) \delta(z_{t+1} = \phi(h_{t+1})) do_{t+1} \right) \max_{a_{t+1}} \mathcal{Q}(z_{t+1}, a_{t+1}) dz_{t+1} \tag{65}$$

$$= \int P(z_{t+1} \mid h_t, a_t) \max_{a_{t+1}} \mathcal{Q}(z_{t+1}, a_{t+1}) dz_{t+1} \tag{66}$$

$$\overset{ZP}{=} \int P(z_{t+1} \mid \phi(h_t), a_t) \max_{a_{t+1}} \mathcal{Q}(z_{t+1}, a_{t+1}) dz_{t+1} = \text{RHS}. \tag{67}$$

□

Lemma 3 (Integral probability metric (Subramanian et al., 2022)). *Given by a function class $\mathcal{F}$, integral probability metric (IPM) between two distributions $\mathbb{P}, \mathbb{Q} \in \Delta(\mathcal{Z})$ is*

$$\mathcal{D}_{\mathcal{F}}(\mathbb{P}, \mathbb{Q}) = \sup_{f \in \mathcal{F}} |\mathbb{E}_{x \sim \mathbb{P}}[f(x)] - \mathbb{E}_{y \sim \mathbb{Q}}[f(y)]|. \tag{68}$$

For any real-valued function g, the following inequality is derived by definition:

$$|\mathbb{E}_{x \sim \mathbb{P}}[g(x)] - \mathbb{E}_{y \sim \mathbb{Q}}[g(y)]| \leq \rho_{\mathcal{F}}(g) \mathcal{D}_{\mathcal{F}}(\mathbb{P}, \mathbb{Q}), \tag{69}$$

where $\rho_{\mathcal{F}}(g) := \inf\{\rho \in \mathbb{R}_+ \mid \rho^{-1} g \in \mathcal{F}\}$ is a Minkowski functional.

Remark 2. Some examples include:

- Total Variance (TV) distance is an IPM defined by $\mathcal{F}_{\text{TV}} = \{f : \|f\|_\infty \leq 1\}$.
- Wasserstein (W) distance is an IPM defined by $\mathcal{F}_{\text{W}} = \{f : \|f\|_L \leq 1\}$.
- KL divergence is not an IPM, but is an upper bound of TV distance by Pinsker's inequality:

$$\mathcal{D}_{\mathcal{F}_{\text{TV}}}(\mathbb{P}, \mathbb{Q}) \leq \sqrt{2 D_{\text{KL}}(\mathbb{P} \,||\, \mathbb{Q})}. \tag{70}$$

Theorem 5 (Approximate version of Thm. 2 (approximate ZP and approximate ϕ_{Q^*} imply approximate ϕ_L)). *Suppose the encoder ϕ satisfies **approximate** ZP (AZP) and we train model-free RL with value parameterized by $\mathcal{Q}(\phi(h), a)$ to **approximate** $Q^*(h, a)$, namely: $\forall t, h_t, a_t$,*

$$\exists P_z : \mathcal{Z} \times \mathcal{A} \to \Delta(\mathcal{Z}), \quad \text{s.t.} \quad \mathcal{D}_{\mathcal{F}}(P(z_{t+1} \mid h_t, a_t), P_z(z_{t+1} \mid \phi(h_t), a_t)) \leq \delta_t, \tag{AZP}$$

$$|Q^*(h_t, a_t) - \mathcal{Q}(\phi(h_t), a_t)| \leq \alpha_t. \tag{Approx. ϕ_{Q^*}}$$

where $\mathcal{D}_{\mathcal{F}}$ is an IPM. Under these conditions, we can construct a latent reward function:

$$\mathcal{R}_z(z_t, a_t) := \begin{cases} \mathcal{Q}(z_t, a_t) & t = T, \\ \mathcal{Q}(z_t, a_t) - \gamma \mathbb{E}_{z_{t+1} \sim P_z(|z_t, a_t)} \left[\max_{a_{t+1}} \mathcal{Q}(z_{t+1}, a_{t+1}) \right] & \text{else}, \end{cases} \tag{71}$$

such that

$$|\mathbb{E}[r_t \mid h_t, a_t] - \mathcal{R}_z(\phi(h_t), a_t)| \leq \epsilon_t, \quad \forall t, h_t, a_t, \tag{ARP}$$

$$\text{where} \quad \epsilon_t = \begin{cases} \alpha_T & t = T, \\ \alpha_t + \gamma(\alpha_{t+1} + \rho_{\mathcal{F}}(\mathcal{V}_{t+1}) \delta_t) & \text{else}, \end{cases} \tag{72}$$

$$\mathcal{V}(z_t) = \max_{a_t} \mathcal{Q}(z_t, a_t), \tag{73}$$

where $\mathcal{V}_{t+1}$ is the latent state-value function $\mathcal{V}$ at step $t+1$.

Proof. For the case of $t = T$ in finite-horizon, ARP holds by the assumption of approx. ϕ_{Q^*}. Now we discuss generic case of t. Recall the reward and latent reward can be rewritten as:

$$\mathbb{E}[r_t \mid h_t, a_t] = Q^*(h_t, a_t) - \gamma \mathbb{E}_{o_{t+1} \sim P(|h_t, a_t)} \left[\max_{a_{t+1}} Q^*(h_{t+1}, a_{t+1}) \right], \tag{74}$$

$$\mathcal{R}_z(z_t, a_t) = \mathcal{Q}(z_t, a_t) - \gamma \mathbb{E}_{z_{t+1} \sim P(|z_t, a_t)} \left[\max_{a_{t+1}} \mathcal{Q}(z_{t+1}, a_{t+1}) \right]. \tag{75}$$

Therefore, the reward gap is upper bounded:

$$|\mathbb{E}[r_t \mid h_t, a_t] - \mathcal{R}_z(\phi(h_t), a_t)| \tag{76}$$

$$\leq |Q^*(h_t, a_t) - \mathcal{Q}(\phi(h_t), a_t)| \tag{77}$$

$$+ \gamma \left| \mathbb{E}_{o_{t+1} \sim P(|h_t, a_t)} \left[\max_{a_{t+1}} Q^*(h_{t+1}, a_{t+1}) \right] - \mathbb{E}_{z_{t+1} \sim P(|\phi(h_t), a_t)} \left[\max_{a_{t+1}} \mathcal{Q}(z_{t+1}, a_{t+1}) \right] \right| \tag{78}$$

$$\leq \alpha_t + \gamma \left| \mathbb{E}_{o_{t+1} \sim P(|h_t, a_t)} \left[\max_{a_{t+1}} Q^*(h_{t+1}, a_{t+1}) - \max_{a_{t+1}} \mathcal{Q}(\phi(h_{t+1}), a_{t+1}) \right] \right| \tag{79}$$

$$+ \gamma \left| \mathbb{E}_{o_{t+1} \sim P(|h_t, a_t)} \left[\max_{a_{t+1}} \mathcal{Q}(\phi(h_{t+1}), a_{t+1}) \right] - \mathbb{E}_{z_{t+1} \sim P(|\phi(h_t), a_t)} \left[\max_{a_{t+1}} \mathcal{Q}(z_{t+1}, a_{t+1}) \right] \right| \tag{80}$$

$$\leq \alpha_t + \gamma \mathbb{E}_{o_{t+1} \sim P(|h_t, a_t)} \left[\max_{a_{t+1}} |Q^*(h_{t+1}, a_{t+1}) - \mathcal{Q}(\phi(h_{t+1}), a_{t+1})| \right] \tag{81}$$

$$+ \gamma \left| \mathbb{E}_{z_{t+1} \sim P(|h_t, a_t)} \left[\max_{a_{t+1}} \mathcal{Q}(z_{t+1}, a_{t+1}) \right] - \mathbb{E}_{z_{t+1} \sim P(|\phi(h_t), a_t)} \left[\max_{a_{t+1}} \mathcal{Q}(z_{t+1}, a_{t+1}) \right] \right| \tag{82}$$

$$\leq \alpha_t + \gamma \alpha_{t+1} + \gamma \left| \mathbb{E}_{z_{t+1} \sim P(|h_t, a_t)} [\mathcal{V}(z_{t+1})] - \mathbb{E}_{z_{t+1} \sim P(|\phi(h_t), a_t)} [\mathcal{V}(z_{t+1})] \right| \tag{83}$$

$$\leq \alpha_t + \gamma(\alpha_{t+1} + \rho_{\mathcal{F}}(\mathcal{V}_{t+1})\delta_t), \tag{84}$$

where Eq. 77 is by triangle inequality, Eq. 79 is by triangle inequality and approx. ϕ_{Q^*}, Eq. 81 is by the maximum-absolute-difference inequality $|\max f(x) - \max g(x)| \leq \max |f(x) - g(x)|$, Eq. 83 is by approx. ϕ_{Q^*}, and Ln. 84 is by the property of IPM (Eq. 69 in Lemma 3) and AZP. □

Remark 3. In the infinite-horizon problem, assume $\delta_t = \delta$ and $\alpha_t = \alpha$ for any t, and $\mathcal{D}_{\mathcal{F}}$ is Wasserstein distance. Furthermore, assume the latent reward $\mathcal{R}_z(z, a)$ is L_r-Lipschitz and the latent transition $P_z(z' \mid z, a)$ is L_p-Lipschitz, then by (Subramanian et al., 2022, Lemma 44), if $\gamma L_p < 1$,

$$\rho_{\mathcal{F}}(\mathcal{V}_{t+1}) = \|\mathcal{V}\|_L \leq \frac{L_r}{1 - \gamma L_p}, \quad \forall t. \tag{85}$$

Thus, the reward difference bound can be rewritten as

$$\epsilon \leq (1 + \gamma)\alpha + \frac{\gamma L_r \delta}{1 - \gamma L_p}. \tag{86}$$

B Objectives and Optimization in Self-Predictive RL

Proof of ***Prop. 1****.* First, we show $\mathcal{L}_{ZP,\ell}(\phi, \theta; h, a) \leq J_\ell(\phi, \theta, \phi; h, a), \forall h, a$.

$$J_\ell(\phi, \theta, \phi; h, a) - \mathcal{L}_{ZP,\ell}(\phi, \theta; h, a) \tag{87}$$

$$= \mathbb{E}_{o' \sim P(|h,a)} \left[\|g_\theta(f_\phi(h), a) - f_\phi(h')\|_2^2 \right] - \|g_\theta(f_\phi(h), a) - \mathbb{E}_{o' \sim P(|h,a)}[f_\phi(h')]\|_2^2 \tag{88}$$

$$= \cancel{\|g_\theta(f_\phi(h), a)\|_2^2} - 2\mathbb{E}_{o'}[\cancel{\langle g_\theta(f_\phi(h), a), f_\phi(h') \rangle}] + \mathbb{E}_{o'} \left[\|f_\phi(h')\|_2^2 \right] \tag{89}$$

$$- \cancel{\|g_\theta(f_\phi(h), a)\|_2^2} + 2\langle g_\theta(f_\phi(h), a), \mathbb{E}_{o'}[f_\phi(h')] \rangle - \|\mathbb{E}_{o'}[f_\phi(h')]\|_2^2 \tag{90}$$

$$= \mathbb{E}_{o'} \left[\|f_\phi(h') - \mathbb{E}_{o'}[f_\phi(h')]\|_2^2 \right] \geq 0. \tag{91}$$

The inner product terms are cancelled due to the fact that inner product is bilinear. The equality holds when $P(o' \mid h, a)$ is deterministic.

Second, the ideal objective using ℓ_2 distance $\mathcal{L}_{ZP,\ell}(\phi, \theta; h, a) = \|g_\theta(f_\phi(h), a) - \mathbb{E}_{o' \sim P(|h,a)}[f_\phi(h')]\|_2^2$ can only lead to EZP condition when reaching optimum of zero. This is because when $g_\theta(f_\phi(h), a) = \mathbb{E}_{o' \sim P(|h,a)}[f_\phi(h')], \forall h, a$, it precisely satisfies the EZP condition. □

Proof of ***Prop. 2****.* The goal is to show $\mathcal{L}_{ZP,D_\mathtt{f}}(\phi, \theta; h, a) \leq J_{D_\mathtt{f}}(\phi, \theta, \phi; h, a), \forall h, a$.

Recall the definition of $\mathtt{f}$-divergence that subsumes forward and reverse KL divergences: $D_\mathtt{f}(Q \mid\mid P) = \int P(x) f\left(\frac{Q(x)}{P(x)}\right) dx$, where $f : [0, \infty) \to \mathbb{R}$ is a convex function.

$$\mathcal{L}_{ZP,D_\mathtt{f}}(\phi, \theta; h, a) = \int \mathbb{P}_\phi(z \mid h) D_\mathtt{f}(\mathbb{P}_\phi(z' \mid h, a) \mid\mid \mathbb{P}_\theta(z' \mid z, a)) dz \tag{92}$$

$$= \iint \mathbb{P}_\phi(z \mid h)\mathbb{P}_\theta(z' \mid z, a) f\left(\frac{\mathbb{E}_{o'}[\mathbb{P}_\phi(z' \mid h')]}{\mathbb{P}_\theta(z' \mid z, a)}\right) dzdz' \tag{93}$$

$$= \iint \mathbb{P}_\phi(z \mid h)\mathbb{P}_\theta(z' \mid z, a) f\left(\mathbb{E}_{o'}\left[\frac{\mathbb{P}_\phi(z' \mid h')}{\mathbb{P}_\theta(z' \mid z, a)}\right]\right) dzdz' \tag{94}$$

$$\leq \iint \mathbb{P}_\phi(z \mid h)\mathbb{P}_\theta(z' \mid z, a)\mathbb{E}_{o'}\left[f\left(\frac{\mathbb{P}_\phi(z' \mid h')}{\mathbb{P}_\theta(z' \mid z, a)}\right)\right] dzdz' \tag{95}$$

$$= \mathbb{E}_{z\sim P_\phi(|h), o'\sim P(|h,a)}\left[\int \mathbb{P}_\theta(z' \mid z, a) f\left(\frac{\mathbb{P}_\phi(z' \mid h')}{\mathbb{P}_\theta(z' \mid z, a)}\right) dz'\right] \tag{96}$$

$$= J_{D_f}(\phi, \theta, \phi; h, a), \tag{97}$$

where we use Jensen's inequality by the convexity of f. The equality holds when $P(o' \mid h, a)$ is deterministic according to Jensen's inequality. □

Discussion on the double sampling issue. The ideal objective Eq. 1 is hard to have an *unbiased* estimate in stochastic environments. This is due to double sampling issue (Baird, 1995) that we do not allow agent to i.i.d. sample twice from transition, *i.e.* $o'_1, o'_2 \sim P(o' \mid h, a)$. To see it more clearly, for example, when $\mathbb{D}$ is forward KL divergence, The ideal objective becomes:

$$\mathcal{L}_{ZP,\text{FKL}}(\phi, \theta; h, a) = \mathbb{E}_{z\sim\mathbb{P}_\phi(|h)}[D_{\text{KL}}(\mathbb{P}_\phi(z' \mid h, a) \,||\, \mathbb{P}_\theta(z' \mid z, a))] \tag{98}$$

$$= \mathbb{E}_{z\sim\mathbb{P}_\phi(|h)}[D_{\text{KL}}(\mathbb{E}_{o'}[\mathbb{P}_\phi(z' \mid h')] \,||\, \mathbb{P}_\theta(z' \mid z, a))] \tag{99}$$

$$= \mathbb{E}_{z\sim\mathbb{P}_\phi(|h)}\left[\int \mathbb{E}_{o'}[\mathbb{P}_\phi(z' \mid h')] \log \frac{\mathbb{E}_{o'}[\mathbb{P}_\phi(z' \mid h')]}{\mathbb{P}_\theta(z' \mid z, a)} dz'\right] \tag{100}$$

$$= \mathbb{E}_{z\sim\mathbb{P}_\phi(|h), o'\sim P(|h,a), z'\sim\mathbb{P}_\phi(|h')}\left[\log \frac{\mathbb{E}_{o'_+}\left[\mathbb{P}_\phi(z' \mid h'_+)\right]}{\mathbb{P}_\theta(z' \mid z, a)}\right], \tag{101}$$

where $o', o'_+ \sim P(|\, h, a)$ are two i.i.d. samples, and $h' = (h, a, o'), h'_+ = (h, a, o'_+)$.

*Proof of **Prop. 3**.* Recall the stop-gradient objective Eq. 2:

$$J := J_\ell(\phi, \theta, \overline{\phi}; h, a) = \mathbb{E}_{o'\sim P(|h,a)}\left[\|g_\theta(f_\phi(h), a) - f_{\overline{\phi}}(h')\|_2^2\right]. \tag{102}$$

The gradients are:

$$\nabla_{\phi,\theta}\mathbb{E}_{h,a}[J] = \mathbb{E}_{h,a}\left[(g_\theta(f_\phi(h), a) - \mathbb{E}_{o'}\left[f_{\overline{\phi}}(h')\right])^\top \nabla_{\phi,\theta} g_\theta(f_\phi(h), a)\right]. \tag{103}$$

When θ, ϕ reaches a stationary point, we have $\nabla_{\phi,\theta}\mathbb{E}_{h,a}[J] = 0$ and thus $\overline{\phi} = \phi$. Therefore, we have *a* stationary point (θ^*, ϕ^*) such that: $g_\theta(f_\phi(h), a) = \mathbb{E}_{o'\sim P(|h,a)}[f_\phi(h')]$, for any h, a, which is the **expected** ZP (EZP) condition. In a deterministic environment, EZP is equivalent to ZP. In a stochastic environment, EZP is to match the expectation (instead of distribution).

However, in online objective, the gradient w.r.t. ϕ contains an extra term:

$$\mathbb{E}_{h,a,o'}\left[(g_\theta(f_\phi(h), a) - f_\phi(h'))^\top \nabla_\phi f_\phi(h')\right]. \tag{104}$$

Thus, when EZP holds, the gradient is zero in deterministic environments, but can be non-zero in stochastic environments.

□

*Proof of **Thm. 3**.* **The setup.** Let $h_{t:-k}$ a vectorization of the recent truncation of history h_t with window size of $k \in \mathbb{N}$, *i.e.* $h_{t:-k} = \text{vec}(a_{t-k}, o_{t-k+1}, \ldots, a_{t-1}, o_t) \in \mathbb{R}^x$,[14] where $x = k(|\mathcal{O}| + |\mathcal{A}|)$. We assume a linear encoder that maps history $h_t \in \mathcal{H}_t$ into z_t:

$$z_t = f_\phi(h_t) := \phi^\top h_{t:-k} \in \mathbb{R}^d, \tag{105}$$

[14]We zero pad a_i and o_i if $i \leq 0$.

where $k \in \mathbb{N}$ is a constant, and the parameters $\phi \in \mathbb{R}^{x \times d}$. In other words, the linear encoder only operates on recent histories of a fixed window size. We assume a linear deterministic latent transition

$$z_{t+1} = g_\theta(z_t, a_t) := \theta_z^\top z_t + \theta_a^\top a_t \in \mathbb{R}^d, \tag{106}$$

where the parameters $\theta_z \in \mathbb{R}^{d \times d}$ and $\theta_a \in \mathbb{R}^{a \times d}$. In fact, the result can be generalized to a non-linear dependence of actions.

The proof. The continuous-time training dynamics of ϕ:

$$\dot{\phi} = -\mathbb{E}_{h_t, a_t}\left[\nabla_\phi J_\ell(\phi, \theta, \overline{\phi}; h_t, a_t)\right] \tag{107}$$

$$= -\mathbb{E}_{h_t, a_t, o_{t+1}}\left[\nabla_\phi \|\theta_z^\top \phi^\top h_{t:-k} + \theta_a^\top a_t - \overline{\phi}^\top h_{t+1:-k}\|_2^2\right] \tag{108}$$

$$= -\mathbb{E}_{h_t, a_t}\left[(\theta_z^\top \phi^\top h_{t:-k} + \theta_a^\top a_t - \mathbb{E}_{o_{t+1}}\left[\overline{\phi}^\top h_{t+1:-k}\right])^\top \nabla_\phi \theta_z^\top \phi^\top h_{t:-k}\right] \tag{109}$$

$$= -\mathbb{E}_{h_t, a_t}\left[h_{t:-k}(\theta_z^\top \phi^\top h_{t:-k} + \theta_a^\top a_t - \mathbb{E}_{o_{t+1}}\left[\overline{\phi}^\top h_{t+1:-k}\right])^\top\right]\theta_z^\top. \tag{110}$$

The gradient of the loss w.r.t. θ_z:

$$\nabla_{\theta_z} \mathbb{E}_{h_t, a_t}\left[J_\ell(\phi, \theta, \overline{\phi}; h_t, a_t)\right] \tag{111}$$

$$= \mathbb{E}_{h_t, a_t}\left[(\theta_z^\top \phi^\top h_{t:-k} + \theta_a^\top a_t - \mathbb{E}_{o_{t+1}}\left[\overline{\phi}^\top h_{t+1:-k}\right])^\top \nabla_{\theta_z}(\theta_z^\top \phi^\top h_{t:-k} + \theta_a^\top a_t)\right] \tag{112}$$

$$= \phi^\top \mathbb{E}_{h_t, a_t}\left[h_{t:-k}(\theta_z^\top \phi^\top h_{t:-k} + \theta_a^\top a_t - \mathbb{E}_{o_{t+1}}\left[\overline{\phi}^\top h_{t+1:-k}\right])^\top\right] \in \mathbb{R}^{d \times d}. \tag{113}$$

Therefore, we have

$$\phi^\top \dot{\phi} = -\nabla_{\theta_z} \mathbb{E}_{h_t, a_t}\left[J_\ell(\phi, \theta, \overline{\phi}; h_t, a_t)\right]\theta_z^\top. \tag{114}$$

Following the practice in (Tang et al., 2022), we assume $\nabla_{\theta_z} \mathbb{E}_{h_t, a_t}\left[J_\ell(\phi, \theta, \overline{\phi}; h_t, a_t)\right] = 0$, *i.e.* θ_z reaches the stationary point of the inner optimization that depends on ϕ, then $\phi^\top \dot{\phi} = 0$. Thus, the training dynamics of $\phi^\top \phi$ is

$$\frac{d(\phi^\top \phi)}{dt} = \dot{\phi}^\top \phi + \phi^\top \dot{\phi} = \dot{\phi}^\top \phi + (\dot{\phi}^\top \phi)^\top = 0. \tag{115}$$

This means that $\phi^\top \phi$ keeps same value during training. □

C Analyzing Prior Works on State and History Representation Learning

In this section, we provide a concise but analytical overview of previous works that learn or approximate self-predictive or observation-predictive representations on states or histories. Please see Lesort et al. (2018) for an early and detailed survey on *state* representation learning.

We focus on the objectives of state or history encoders in their value functions. For each work discussed, we present a summary of the conditions that their encoders aim to satisfy or approximate at the beginning of each paragraph. In cases where multiple encoder objectives are proposed, we select the one employed in their primary experiments for our discussion. In particular, we list the exact objectives they aim to optimize, which might be redundant for *exact* conditions. For example, multi-step RP can be implied by RP + ZP by Prop. 11 (or ϕ_{Q^*} + ZP by Thm. 2), and multi-step ZP can be implied by ZP by Prop. 10.

C.1 Self-Predictive Representations

CRAR (François-Lavet et al., 2019): ϕ_{Q^*} + RP + ZP with online ℓ_2 + regularization. Combined reinforcement via abstract representations (CRAR) is designed to learn self-predictive representations in MDPs. It incorporates RP and ZP auxiliary losses into the end-to-end RL objective. They assume the deterministic case (for the encoder, transition and latent transition), thus using ℓ_2 objective is sufficient (Prop. 1). They use online ZP target and observe the representation collapse when the reward signals are scarce. To prevent this issue, they introduce regularization terms into the encoder objective. These terms minimize $\mathbb{E}_{s_1, s_2}[\exp(-\|\phi(s_1) - \phi(s_2)\|_2)] + \mathbb{E}_s\left[\max(\|\phi(s)\|_\infty^2 - 1, 0)\right]$, where s_1, s_2, s are samples from the state space. These terms, similar to entropy maximization, encourage diversity within the latent space.

DeepMDP (Gelada et al., 2019): ϕ_{Q^*} + RP + ZP with online ℓ_2. DeepMDP aims to learn state representations that match RP and ZP. In their experiments, they assume deterministic case, resulting in dirac distributions $\mathbb{P}_\phi(z' \mid s')$ and $\mathbb{P}_\theta(z' \mid z, a)$. Although they use the Wasserstein distance, it reduces to ℓ_2 distance for two dirac distributions. They use an online target in ZP loss. In their toy DonutWorld task, they try phased training with RP + ZP, but the agent tends to be trapped in a local minimum of zero ZP. Then they try ϕ_{Q^*} + RP + ZP in Atari by training RP + ZP as an auxiliary task of a distributional RL baseline, outperforming the baseline in their main result. They also find that ϕ_{Q^*} + RP + ZP is comparable to ϕ_{Q^*} + ZP, aligned with our theoretical prediction based on Thm. 2. They also try phased training in Atari and find that RP + ZP performs poorly, while RP + ZP + OR yields good results.

SPR (Schwarzer et al., 2020): ϕ_{Q^*} + multi-step ZP with EMA cos. Self-Predictive Representations (SPR) improves the ZP objective in DeepMDP. They use a special kind of ℓ_2 loss (*i.e.* cos distance) to bound the loss scale, and use an EMA target. They use multi-step prediction loss to learn the condition:

$$P(z_{t+1:t+k} \mid s_t, a_{t:t+k-1}) = P(z_{t+1:t+k} \mid \phi(s_t), a_{t:t+k-1}), \tag{116}$$

where $k = 5$ in their experiments. In addition, to reduce the large latent space generated by CNNs, they use a linear projection of the latent states to satisfy ZP.

DBC (Zhang et al., 2020): ϕ_{Q^*} + RP + stronger ZP with detached FKL. Deep Bisimulation for Control (DBC) trains the state encoder ϕ with several auxiliary losses, including RP and ZP. The ZP loss uses a forward KL objective with a detached target. Their main contribution is the introducation of the bisimulation metric (Ferns et al., 2004) into state representation learning: for any $s_i, s_j \in \mathcal{S}$ and $a_i, a_j \in \mathcal{A}$,

$$\|\phi(s_i) - \phi(s_j)\|_1 = |R(s_i, a_i) - R(s_j, a_j)| + \gamma W(\mathbb{P}_\theta(z' \mid \phi(s_i), a_i), \mathbb{P}_\theta(z' \mid \phi(s_j), a_j)), \tag{metric}$$

where W is Wasserstein distance and $\mathbb{P}_\theta$ is modeled as a Gaussian. The metric condition enforces the latent space to be structured with a ℓ_1 metric. They train ϕ satisfying the metric condition by minimizing the mean square error on it as another auxiliary loss. This leads to a stronger ZP condition.

PBL (Guo et al., 2020): ϕ_{Q^*} + indirect multi-step ZP. Predictions of Bootstrapped Latents (PBL) designs two auxiliary losses, reverse prediction and forward prediction, for their history encoder ϕ, transition model θ, observation encoder f, and projector g:

$$\min_{f,g} \mathbb{E}_h \left[\|g(f(o)) - \phi(h)\|_2^2 \right], \tag{Reverse}$$

$$\min_{\phi,\theta} \mathbb{E}_{h,a,o'} \left[\|\theta(\phi(h), a) - f(o')\|_2^2 \right]. \tag{Forward}$$

To understand their connection with ZP, assume the two losses reach zero with $\phi(h) = g(f(o))$ and $\theta(\phi(h), a) = \mathbb{E}_{o' \sim P(|h,a)}[f(o')]$ for any h, a, although in theory this may be unrealizable. Furthermore, assume deterministic transition, then

$$g(\theta(\phi(h), a)) = g(f(o')) = \phi(h'). \tag{117}$$

Therefore, in deterministic environments, reverse and forward prediction together is equivalent to ZP if they reach the optimum. They also adopt multi-step version of their loss with a horizon of 20. While forward and reverse prediction both appear critical in this work, the follow-up work BYOL-explore (Guo et al., 2022) removes reverse prediction.

Successor Representations and Features (Barreto et al., 2017; Lehnert & Littman, 2020): ϕ_{Q^*} + RP + weak ZP. Here, we introduce successor features (SF) with our notation. Suppose the expected reward function can be computed as

$$\mathbb{E}[r \mid s, a] = g(\phi(s), a)^\top w, \quad \forall s, a, \tag{118}$$

where $\phi : \mathcal{S} \to \mathcal{Z}$ is a state encoder and $g : \mathcal{Z} \times \mathcal{A} \to \mathbb{R}^d$ is called state-action feature extractor, and $w \in \mathbb{R}^d$ are weights[15]. In our notation, Eq. 118 is RP condition for ϕ.

[15]Although it is linear w.r.t. w, it can recover any reward function, *e.g.* when $\phi(s) = s$ and $g(s, a)_i = \mathbb{E}[r \mid s, a]$ for some i.

As a special case, in tabular MDPs with finite state and action spaces with state-dependent reward $R(s)$, let $\phi(s) \in \{0,1\}^{|\mathcal{S}|}$ be one-hot state representation, and let $g(\phi(s), a) = \phi(s)$ and weight $w_s = \mathbb{E}[r \mid s]$, this satisfies Eq. 118. This special case is known as **successor representation** (SR) setting (Dayan, 1993). In deep SR (Kulkarni et al., 2016; Lehnert & Littman, 2020), they allow learning ϕ with assuming $g(\phi(s), a) = \phi(s)$.

The Q-value function of a policy π can be rewritten as

$$Q^\pi(s,a) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid S_0 = s, A_0 = a\right] \tag{119}$$

$$= \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t g(\phi(s_t), a_t)^\top w \mid S_0 = s, A_0 = a\right] \tag{120}$$

$$= \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t g(\phi(s_t), a_t) \mid S_0 = s, A_0 = a\right]^\top w \tag{121}$$

$$:= \psi^\pi(s,a)^\top w, \tag{122}$$

where $\psi^\pi(s,a)$ is called successor features (Barreto et al., 2017), a geometric sum of future $g(\phi(s), a)$. Although ψ^π can belong to any function class, following deep SR (Kulkarni et al., 2016; Lehnert & Littman, 2020), we assume it is parametrized by the state encoder as $\psi^\pi(s,a) = f^\pi(\phi(s), a)$ where $f^\pi : \mathcal{Z} \times \mathcal{A} \to \mathbb{R}^d$. Then, by plugging Eq. 122 in Bellman equation $Q^\pi(s,a) = \mathbb{E}_{s',a' \sim \pi}[R(s,a) + \gamma Q^\pi(s',a')]$, we have

$$f^\pi(\phi(s), a) = g(\phi(s), a) + \gamma \mathbb{E}_{s',a' \sim \pi}[f^\pi(\phi(s'), a')]. \tag{123}$$

Therefore, Eq. 123 can be viewed as a **weak** version of ZP, because given any current latent state and action pair $(\phi(s), a)$, Eq. 123 can predict the expectation of some function of next latent state $\phi(s')$. ZP can imply Eq. 123 because it can predict exactly the distribution of next latent state.

With a combination of RP (Eq. 118), ϕ_{Q^*} (implied by Eq. 123 when π is optimal), and a weak version of ZP, we show that the state encoder that successor features learn, belongs to a weak version of ϕ_L.

As a special case, in Linear Successor Feature Model (LSFM) (Lehnert & Littman, 2020, Theorem 2), they show that SF is **exactly** the bisimulation (ϕ_L) under several assumptions: finite action and latent space, the successor features $f^\pi(z,a) = F_a z$ is a linear function, and the policy $\pi : \mathcal{Z} \to \Delta(\mathcal{A})$ conditions on latent space. However, here we point it out that with the assumptions above implies EZP (not necessarily ZP), thus, still a **weak** version of bisimulation.

Following Lehnert & Littman (2020), assume the finite latent space is composed of one-hot vectors: $\mathcal{Z} = \{e_1, e_2, \ldots, e_n\}$, we can construct a matrix $F^\pi \in \mathbb{R}^{d \times n}$ with each column $F^\pi(i) = \mathbb{E}_{a \sim \pi(|e_i)}[F_a e_i]$.

$$\frac{1}{\gamma}(f^\pi(\phi(s), a) - g(\phi(s), a)) = \mathbb{E}_{s',a' \sim \pi}[f^\pi(\phi(s'), a')] \tag{124}$$

$$= \mathbb{E}_{s' \sim P(|s,a), a' \sim \pi(|\phi(s'))}[F_{a'} \phi(s')] \tag{125}$$

$$= \mathbb{E}_{s' \sim P(|s,a)}[F^\pi \phi(s')] = F^\pi \mathbb{E}_{s' \sim P(|s,a)}[\phi(s')]. \tag{126}$$

By (Lehnert & Littman, 2020, Lemma 4), F^π is invertible, thus there exists a function $J : \mathcal{Z} \times \mathcal{A} \to \mathcal{Z}$ such that $J(\phi(s), a) = \mathbb{E}_{s' \sim P(|s,a)}[\phi(s')]$, *i.e.*, EZP holds.

EfficientZero (Ye et al., 2021): ϕ_{Q^*} + RP + multi-step ZP with detached cos. EfficientZero improves MuZero (Schrittwieser et al., 2020) by introducing ZP loss as one of their main contributions. We consider it especially crucial to planning algorithms because ZP enforces the latent model to be accurate. Similar to SPR (Schwarzer et al., 2020), they use 5-step cos objective with a projection on latent states, and add image data augmentation for visual RL tasks.

RPC (Eysenbach et al., 2021): ϕ_{π^*} + ZP with online forward KL. From the perspective of information compression, robust predictive control (RPC) aims to jointly learn the encoder of policy $\mathbb{P}_\phi(z \mid s)$ and the latent policy $\pi_z(a \mid z)$ in MDPs. The policy $\pi(a \mid s)$ is not only maximizing return,

but also imposed a constraint on $\mathbb{E}_\pi[I(s_{1:\infty}; z_{1:\infty})] \leq C$ where $C > 0$ is a predefined constant. By applying variational information bottleneck, this constraint induces the algorithm RPC to maximize the following objective w.r.t. ϕ and θ (see their Eq. 6):

$$\mathcal{L}(\phi, \theta; s_t, a_t) = \mathbb{E}_{z_t \sim \mathbb{P}_\phi(|s_t), s_{t+1} \sim P(|s_t, a_t), z_{t+1} \sim \mathbb{P}_\phi(|s_{t+1})} \left[\log \frac{\mathbb{P}_\theta(z_{t+1} \mid z_t, a_t)}{\mathbb{P}_\phi(z_{t+1} \mid s_{t+1})} \right] \tag{127}$$

$$= -\mathbb{E}_{z_t \sim \mathbb{P}_\phi(|s_t), s_{t+1} \sim P(|s_t, a_t)} [D_{\mathrm{KL}}(\mathbb{P}_\phi(z_{t+1} \mid s_{t+1}) \,||\, \mathbb{P}_\theta(z_{t+1} \mid z_t, a_t))] \tag{128}$$

which is exactly the practical forward KL objective Eq. 3. In practice, the authors formulate it as constrained optimization and use gradient descent-ascent to update the encoder and Lagrange multiplier. In addition, they also use this objective as an intrinsic reward to regularize the latent policy's reward-maximizing objective. It is worth noting that while RPC aims to learn the ZP condition along with reward maximization, it does not explicitly learn representations to fulfill the RP or ϕ_{Q^*} conditions. As a result, we can consider it as an approach that approximates self-predictive representations.

ALM (Ghugare et al., 2022): ϕ_{Q^*} + multi-step ZP with EMA reverse KL. Aligned Latent Models (ALM) is based on variational inference, and aims to learn the latent model $\mathbb{P}_\theta(z' \mid z, a)$, the state encoder $\mathbb{P}_\phi(z \mid s)$ and the latent policy $\pi_z(z)$ to jointly maximize the lower bound of the expected return. The objective of their encoder includes maximizing the return and ZP loss, instantiated as 3-step reverse KL with an EMA target. Specifically, given a tuple of (s, a, s'), the 1-step objective for their encoder is computed as

$$\min_\phi \mathbb{E}_{z \sim \mathbb{P}_\phi(|s)} \left[-R_z(z, a) + D_{\mathrm{KL}}(\mathbb{P}_\theta(z' \mid z, a) \,||\, \mathbb{P}_\phi(z' \mid s')) - \mathbb{E}_{z' \sim \mathbb{P}_\theta(|z,a)} \left[Q^\pi(z', \pi_z(\overline{z'})) \right] \right], \tag{129}$$

where $R_z(z, a)$ is the latent reward, learned by the RP condition (with ϕ detached), and $\overline{z'}$ indicates stop-gradient. With the latent reward and also their intrinsic rewards, they perform SVG algorithm (Heess et al., 2015) for policy optimization with a planning horizon of 3 steps. We provide a detailed description of ALM and its variants in Sec. E.2.

AIS (Subramanian et al., 2022): RP + ZP with detached ℓ_2 or forward KL in their approach, while RP + OP with detached ℓ_2 in their experiments. Approximate Information States (AIS) adopts a phased training framework where the history encoder ϕ learns from RP instead of maximizing returns. In their approach section (Subramanian et al., 2022, Sec. 6.1.2), they propose using MMD with ℓ_2 distance-based kernel k_d to learn ZP, and detach the target. The distance-based kernel (Sejdinovic et al., 2013) takes a pair of latent states $z_1, z_2 \in \mathcal{Z}$ as inputs, and is defined as $k_d(z_1, z_2) = \frac{1}{2}(d(z_0, z_1) + d(z_0, z_2) - d(z_1, z_2))$ where $z_0 \in \mathcal{Z}$ is arbitrary. In this case, $d(z_1, z_2) = \|z_1 - z_2\|_2^2$ is ℓ_2 distance.

Let $f_\phi(h)$ be the deterministic encoder, $\mathbb{P}_\theta(z' \mid f_\phi(h), a) := \mathbb{P}_{\phi,\theta}$ be the predicted next latent distribution, and $\mathbb{Q}_\phi(z' \mid h, a)$ be real next latent distribution. The MMD with k_d can be reduced to ℓ_2 distance between the expectations of two distributions:

$$\mathrm{MMD}^2_{k_d}(\mathbb{P}_{\phi,\theta}, \mathbb{Q}_\phi; h, a) \tag{130}$$

$$= -\mathbb{E}_{z'_1, z'_2 \sim \mathbb{P}_{\phi,\theta}}[d(z'_1, z'_2)] + 2\mathbb{E}_{z'_1 \sim \mathbb{P}_{\phi,\theta}, z'_2 \sim \mathbb{Q}_\phi}[d(z'_1, z'_2)] - \mathbb{E}_{z'_1, z'_2 \sim \mathbb{Q}_\phi}[d(z'_1, z'_2)] \tag{131}$$

$$= -\mathbb{E}_{z'_1, z'_2 \sim \mathbb{P}_{\phi,\theta}} \left[\|z'_1 - z'_2\|_2^2 \right] + 2\mathbb{E}_{z'_1 \sim \mathbb{P}_{\phi,\theta}, z'_2 \sim \mathbb{Q}_\phi} \left[\|z'_1 - z'_2\|_2^2 \right] - \mathbb{E}_{z'_1, z'_2 \sim \mathbb{Q}_\phi} \left[\|z'_1 - z'_2\|_2^2 \right] \tag{132}$$

$$= 2\|\mathbb{E}_{z' \sim \mathbb{P}_{\phi,\theta}}[z' \mid h, a] - \mathbb{E}_{z' \sim \mathbb{Q}_\phi}[z' \mid h, a]\|_2^2. \tag{133}$$

Therefore, the MMD objective can be viewed as ZP with ℓ_2 distance (*i.e.*, EZP). They also propose forward KL to instantiate ZP loss. Nevertheless, AIS (Subramanian et al., 2022) do not show experiment results on learning ZP. Instead, they and the follow-up works (Patil et al., 2022; Seyedsalehi et al., 2023) implement AIS by learning OP loss with MMD objectives, resulting in learning *observation-predictive* representations. Another follow-up work, Discrete AIS (Yang et al., 2022), learns ZP loss with ℓ_2 objective in a discrete latent space, so that they can apply value iteration.

TD-MPC (Hansen et al., 2022): ϕ_{Q^*} + RP + multi-step ZP with EMA ℓ_2. Temporal Difference learning for Model Predictive Control (TD-MPC) uses a planning horizon of 5 for the encoder objective and the latent value objective with TD learning. TD-MPC also uses MPC for action selection during inference. They find that learning ZP works better than learning OR or not learning ZP in the DM Control suite.

TCRL (Zhao et al., 2023): RP + multi-step ZP with EMA cos. Temporal consistency reinforcement learning (TCRL) simplifies TD-MPC (Hansen et al., 2022) by removing the planning component, replacing ℓ_2 loss with cos loss, and detaching the encoder parameters during value function learning. They validate their approach on the state-based DM Control suite. Although the paper refers to TCRL as *minimalist* for learning representations, it is worth noting that TCRL is more complicated than our approach, as it still requires reward prediction and multi-step prediction.

C.2 Observation-Predictive Representations

PSR (Littman et al., 2001) and belief trajectory equivalence (Castro et al., 2009): Rec + multi-step OP and RP. Predictive State Representation (PSR) aims to learn a history encoder ϕ and transition model P_O such that

$$P(o_{t+1:t+k} \mid h_t, a_{t:t+k-1}) = P_O(o_{t+1:t+k} \mid \phi(h_t), a_{t:t+k-1}), \quad \forall h, a, o, \tag{134}$$

which implies multi-step OP (defined in Prop. 12) in POMDPs. The original PSR uses linear transition models. Follow-up work on PSRs (James et al., 2004) and belief trajectory equivalence introduce multi-step RP to PSR. In Castro et al. (2009), they show that single-step OP and RP do not necessarily imply multi-step OP and RP in POMDPs, summarized in Prop. 12. In this sense, PSR is a stronger notion of belief abstraction.

Causal state representations (Zhang et al., 2019): Rec + OP + RP. This work connects observation-predictive representations in POMDPs with causal state models in computational mechanics (Shalizi & Crutchfield, 2001). Specifically, they show that belief trajectory equivalence (Rec + multi-step OP and RP) (Castro et al., 2009) implies a causal state of a stochastic process, where RP means reward *distribution* prediction. The resulting abstract MDP is a causal state model or an ϵ-machine, generating minimal sufficient representations for predicting future observations. In the implementation, they train a deterministic RNN encoder and a deterministic transition model to satisfy OP and RP conditions, and also train a latent Q-value function using Q-learning by freezing encoder parameters. Optionally, they also train a discretizer on the latent space in finite POMDPs.

Belief-Based Methods (Hafner et al., 2019; 2020b; Han et al., 2020; Lee et al., 2020): RP + OR + ZP with online forward KL. As a major approach to solving POMDPs, belief-based methods extends belief MDPs (Kaelbling et al., 1998) to deep RL through variational inference, deriving the encoder objective as ELBO. Let the latent variables are $z_{1:T}$, the world model $p(o_{1:T}, r_{1:T} \mid a_{1:T})$, and the posterior are $q(z_{1:T} \mid o_{1:T}, a_{1:T})$ with the factorization:

$$p(z_{1:T+1}, o_{1:T+1}, r_{1:T} \mid a_{1:T}) = p(z_1)p(o_1 \mid z_1) \prod_{t=1}^{T} p(r_t \mid z_t, a_t)p(z_{t+1} \mid z_t, a_t)p(o_{t+1} \mid z_{t+1}), \tag{135}$$

$$q(z_{1:T+1} \mid h_{T+1}) = \prod_{t=0}^{T} q(z_{t+1} \mid h_{t+1}) = \prod_{t=0}^{T} q(z_{t+1} \mid z_t, a_t, o_{t+1}), \tag{136}$$

where $h_{t+1} = (h_t, a_t, o_{t+1})$ in our notation. The log-likelihood has a lower bound:

$$\mathbb{E}_{h_{T+1}, r_{1:T}}[\log p_\theta(o_{1:T+1}, r_{1:T} \mid a_{1:T})] \tag{137}$$

$$= \mathbb{E}_{h_{T+1}, r_{1:T}}\left[\log \mathbb{E}_{q(z_{1:T+1} \mid h_{T+1})}\left[\frac{p(z_{1:T+1}, o_{1:T+1}, r_{1:T} \mid a_{1:T})}{q(z_{1:T+1} \mid h_{T+1})}\right]\right] \tag{138}$$

$$\geq \mathbb{E}_{h_{T+1}, r_{1:T}, z_{1:T} \sim q(\cdot \mid h_{T+1})}\left[\log \frac{p(z_{1:T}, o_{1:T+1}, r_{1:T} \mid a_{1:T})}{q(z_{1:T+1} \mid h_{T+1})}\right] \tag{139}$$

$$= \mathbb{E}_{h_{T+1}, r_{1:T}, z_{1:T+1} \sim q(h_{T+1})}\left[\sum_{t=0}^{T} \underbrace{\log p(o_{t+1} \mid z_{t+1})}_{(1)} + \underbrace{\log p(r_t \mid z_t, a_t)}_{(2)} - \underbrace{\log \frac{q(z_{t+1} \mid h_{t+1})}{p(z_{t+1} \mid z_t, a_t)}}_{(3)}\right]. \tag{140}$$

When p, q are trained to optimal, the first term becomes OR condition and the second term becomes reward distribution matching that implies RP. The third term with expectation can be written as $\mathbb{E}_{z_t,h_{t+1}}[D_{\mathrm{KL}}(q(z_{t+1} \mid h_{t+1}) \,||\, p(z_{t+1} \mid z_t, a_t))]$, which is exactly our practical forward KL objective Eq. 3 to learn ZP. From our relation graph (Fig. 1; Prop. 8), ZP + OR imply OP, thus belief-based methods aim to approximate observation-predictive representation (RP + OP). Normally, they use an online target in forward KL, because they have OR signals that can help prevent representational collapse. They also train encoders without maximizing returns.

We can also build the connections between OR and RP objectives and maximizing mutual information. Let $P(o, z)$ be the marginal joint distribution of observation and latent state at the same time-step, where $P(o', z') = \int P(o', z', h, a) dh da = \int P(h, a) P(o' \mid h, a) P(z' \mid h') dh da$. Consider,

$$\mathbb{I}(o'; z') = \mathbb{E}_{o',z' \sim P(o',z')} \left[\log \frac{P(o', z')}{P(o')P(z')} \right] \tag{141}$$

$$= \mathbb{E}_{o',z' \sim P(o',z')} \left[\log \frac{P(o' \mid z')}{P(o')} \right] \tag{142}$$

$$= \mathbb{E}_{o',z' \sim P(o',z')} [\log P(o' \mid z')] + \mathbb{H}(P(o')) \tag{143}$$

$$= \mathbb{E}_{h,a \sim P(h,a), o' \sim P(|h,a), z' \sim P(|h')} [\log P(o' \mid z')] + \mathbb{H}(P(o')). \tag{144}$$

Since the entropy term is independent of latent states, the OR objective in belief-based methods is **exactly** maximizing the $\mathbb{I}(o; z)$. Similarly, the RP objective in belief-based methods is exactly maximizing $\mathbb{I}(r; z)$.

OFENet (Ota et al., 2020): ϕ_{Q^*} + OP. Online Feature Extractor Network (OFENet) trains the state encoder using an auxiliary task of OP loss with ℓ_2 distance. This is perhaps the most related algorithm to our Algo. 1 for learning Φ_O. They show strong performance of their approach over model-free baseline in standard MuJoCo benchmark. Follow-up work (Lange et al., 2023) empirically find that ϕ_{Q^*} + RP slightly improves up model-free RL, but much worse than ϕ_{Q^*} + OP in MuJoCo benchmark.

SAC-AE (Yarats et al., 2021): ϕ_{Q^*} + OR. Soft Actor-Critic with AutoEncoder (SAC-AE) trains the state encoder with an auxliary task of OR loss with forward KL and also ℓ_2-regularization. They detach the state encoder in policy objective. As in MDPs, OR implies OP (Prop. 9), SAC-AE also approximates observation-predictive representation.

C.3 Other Related Representations

UNREAL (Jaderberg et al., 2016), Loss is its own Reward (Shelhamer et al., 2016). These works make early attempts at auxiliary task design for RL. UNREAL trains recurrent A3C agent with several auxiliary tasks, including reward prediction (RP), pixel control and value function replay. Loss is its own Reward trains A3C agent with several auxiliary tasks, including reward prediction (RP), observation reconstruction (OR), inverse dynamics, and a proxy of forward dynamics (OP) that finds the corrupted observation from a time series. Among them, inverse dynamics condition in MDPs is that

$$\exists P_{\text{inv}} : \mathcal{Z} \times \mathcal{Z} \to \Delta(\mathcal{A}), \quad s.t. \quad P_{\text{inv}}(a \mid \phi(s), \phi(s')) = P(a \mid s, s'), \quad \forall s, a, s', \tag{145}$$

but this condition does not direct relation with forward dynamics (OP).

VPN (Oh et al., 2017), MuZero (Schrittwieser et al., 2020): ϕ_{Q^*} + RP. From Thm. 2, we know that ϕ_{Q^*} + RP is implied by ϕ_{Q^*} + ZP, thus this representation lies between ϕ_{Q^*} and ϕ_L. Both VPN and MuZero learn the shared state encoder and latent model from maximizing the return and predicting rewards. Their policies are learned by the MCTS algorithm.

E2C (Watter et al., 2015) and World Model (Ha & Schmidhuber, 2018): ZP + OR. They are similar to belief-based methods, but remove the reward prediction loss from the encoder objective. Instead, reward signals are only accessible to latent policies or values.

Contrastive representation learning in RL (CURL (Laskin et al., 2020), DRIML (Mazoure et al., 2020), ContraBAR (Choshen & Tamar, 2023)): ϕ_{Q^*} (RP) + weak OP (OR). CURL (ϕ_{Q^*} + weak OR) introduces contrastive learning using the infoNCE objective (Oord et al., 2018) as an auxiliary task in MDPs. InfoNCE between positive and negative examples is shown to be a lower bound of mutual information between input and latent state variables (Poole et al., 2019). In MDPs, it is a lower bound of $\mathbb{I}(s;z)$, which corresponds to OR objectives Eq. 141. Therefore, CURL can be interpreted as maximizing a lower bound of OR.

DRIML (ϕ_{Q^*} + weak OP) proposes an auxiliary task named InfoMax in MDPs. In its single-step prediction variant, InfoMax maximizes the lower bound of $\mathbb{I}(z';z,a)$ via the infoNCE objective. Similar to the analysis (Rakelly et al., 2021), by data processing inequality:

$$\mathbb{I}(z';z,a) \leq \mathbb{I}(z';s,a) \leq \mathbb{I}(s';s,a), \tag{146}$$

$$\mathbb{I}(z';z,a) \leq \mathbb{I}(s';z,a) \leq \mathbb{I}(s';s,a). \tag{147}$$

When all equalities hold (*e.g.* ϕ satisfies OR), these imply $z' \perp\!\!\!\perp s,a \mid z,a$ (ZP) and $s' \perp\!\!\!\perp s,a \mid z,a$ (OP).

ContraBAR (weak RP and weak OP) introduces infoNCE objectives to meta-RL, which requires incorporating reward signals into observations when viewed as POMDPs (Ni et al., 2022). Similar to DRIML, in its single-step prediction variant, the objective is to maximize the lower bound of mutual information of $\mathbb{I}(z';z,a)$ where z is a joint representation of state s and reward r. As shown in the ContraBAR paper (Choshen & Tamar, 2023, Theorem 4.3), under certain optimality condition, the objective can lead to learning RP and OP conditions.

Learning Markov State Abstraction (Allen et al., 2021): ϕ_{Q^*} + ZM. From Prop. 6, we know that ZM is implied by ZP, thus representation lies between ϕ_{Q^*} and ϕ_L. They show that ZM can be implied by inverse dynamics and density ratio matching in MDPs. Thus, they train on these two objectives as auxiliary losses.

MICo (Castro et al., 2021): ϕ_{Q^*} + metric. With a state encoder ϕ, matching under Independent Coupling (MICo) defines a distance metric U_ϕ in the state space. For any pair of states $x, y \in \mathcal{S}$,

$$U_\phi(x,y) = |r^\pi_x - r^\pi_y| + \gamma \mathbb{E}_{x' \sim P^\pi_x, y' \sim P^\pi_y}[U_\phi(x',y')], \tag{metric}$$

where $r^\pi_x = \mathbb{E}_{a\sim\pi(|x)}[R(x,a)]$ and $P^\pi_x(x' \mid x) = \mathbb{E}_{a\sim\pi(|x)}[P(x' \mid x,a)]$. The metric U_ϕ is parameterized with

$$U_\phi(x,y) = \frac{1}{2}(\|\phi(x)\|_2^2 + \|\phi(y)\|_2^2) + \beta \arctan(\sqrt{1-\cos(\phi(x),\phi(y))^2}, \cos(\phi(x),\phi(y))). \tag{148}$$

They learn the MICo metric by an auxiliary loss using mean squared error.

Denoised MDPs (Wang et al., 2022): OR + RP + ZP in a factorized latent space (implying ϕ_{π^*}, not ϕ_{Q^*}). This work aims to learn a state abstraction that ignores components that are either reward-irrelevant or uncontrollable. Such an abstraction can retain the optimal policy while not necessarily preserving optimal value functions. Consequently, denoised MDPs can be conceptualized as approximating ϕ_{π^*}. Technically, the authors postulate that the latent state of an MDP is composed of elements (x,y,z) where the transition in y is independent of actions. Additionally, the reward function $r(s)$, independent of z, is decomposed into $r_x(x)$ and $r_y(y)$. Thus, the optimal policy (though not its value) can only depend on the latent state component x.

In practice, they introduce variational objectives to learn the encoder $p(x,y,z \mid s)$ with observation reconstruction (OR), reward prediction (RP), and next latent state prediction (ZP) using online forward KL divergences. The structure of the latent state space helps the partial encoder $p(x \mid s)$ to gravitate towards ϕ_{π^*} abstraction, despite the absence of a theoretical guarantee. Finally, they use model-free RL to optimize a policy on the latent x space.

TD7 (Fujimoto et al., 2023): ZP with detached ℓ_2. TD7 algorithm is introduced for addressing MDPs and evaluated on the MuJoCo benchmark. TD7 learns a state encoder using ZP loss with detaching the next latent states, which performs better than EMA version. They use ℓ_2 loss and normalize latent states by average ℓ_1 norm, which performs better than cos loss and other normalization

methods. They find that training with ZP loss only is slightly better than training with ZP + RP, and much better than end-to-end training (ZP + ϕ^*_Q). Lastly, it is noteworthy that in TD7, the critic not only takes a state s and an action a as inputs, but also the latent state $f_\phi(s)$ and the predicted next latent state $g_\theta(f_\phi(s), a)$, which is named as *state-action embedding* in the paper.

D Additional Discussion

D.1 Motivating Our Hypotheses

Here we provide our motivation for our hypotheses shown in Sec. 5.

- **Motivating the sample efficiency hypothesis.** The performance of deep RL algorithms is notably influenced by task structure, and no single algorithm consistently outperforms others across all tasks (Wang et al., 2019; Li et al., 2022; Ni et al., 2022). Common wisdom suggests that certain algorithms excel in specific types of tasks (Mohan et al., 2023). For instance, self-predictive representations are often effective in distracting tasks (Zhang et al., 2020; Zhao et al., 2023), while observation-predictive representations typically perform well in sparse-reward scenarios (Zintgraf et al., 2021; Zhang et al., 2021). However, these methods often incorporate additional complexities like intrinsic rewards and metric learning or are primarily evaluated in pixel-based tasks.

  Given these considerations, we propose the use of our minimalist algorithm as a tool to focus solely on the impact of representation learning in vector-based tasks. This approach aims to provide a clearer understanding of how different representation learning strategies affect sample efficiency in various task structures (including popular standard benchmarks), without the confounding factors present in more complex algorithms or environments.
- **Motivating the distraction hypothesis.** The belief that algorithms predicting observations tend to underperform in distracting tasks is supported by several studies (Zhang et al., 2020; Okada & Taniguchi, 2021; Fu et al., 2021; Deng et al., 2022). The challenge arises from the need for these models to predict every detail of observations, including irrelevant features, which can be extremely difficult due to randomness and high dimensionality. Similar to the motivation in our sample efficiency hypothesis, prior works primarily focus on complex algorithms evaluated in pixel-based tasks, often with real-world video backgrounds as distractors.

  Considering these, we propose a shift towards studying the impact of distractions using a minimalist algorithm in simpler, configurable environments. This approach aims to isolate and understand the specific effects of distracting elements in tasks, providing a more straightforward and controlled setting for analysis.
- **Motivating the end-to-end hypothesis.** According to our Thm. 2, learning an encoder end-to-end with the auxiliary task of ZP can implicitly learn the reward prediction conditioned on its optimality, potentially making it comparable to the phased learning. However, this is a theoretical prediction and may not necessarily translate to practical scenarios, particularly considering that RL agents rarely achieve global optima. On the other hand, prior works (Schwarzer et al., 2020; Ghugare et al., 2022) have shown the success of the end-to-end learning, but these algorithms incorporate other moving components (multi-step prediction, intrinsic rewards) and are not directly applicable to POMDPs.

  These limitations underscore the importance of empirically testing whether the benefits of the end-to-end learning extend to POMDPs when employing a minimalist approach in representation learning.
- **Motivating the ZP objective hypothesis.** Our Prop. 1 and Prop. 2 suggest that it suffices to use ℓ_2 objective in deterministic tasks while KL divergences might be more effective in stochastic ones. However, these are theoretical assumptions that do not fully account for the complexities of the learning process. Additionally, most existing research tends to focus on a single objective type in deterministic settings (as summarized in Table 1), leaving the performance of alternative objectives, particularly in stochastic tasks, largely unexplored. A notable exception is AIS (Subramanian et al., 2022) which discusses various ZP objectives but lacks practical evaluation on them.

  These gaps in the literature motivate us to undertake a thorough comparison of these ZP objectives in practical settings.
- **Motivating the ZP stop-gradient hypothesis.** Our Thm. 3 suggests that applying stop-gradient to ZP targets could help mitigate representational collapse. However, this prediction is based on linear

models without incorporating RL loss, which is a significant departure from deep RL scenarios. While most prior studies focus on one type of ZP target without delving into collapse issues (as summarized in Table 1), SPR (Schwarzer et al., 2020) is an exception, comparing online and EMA encoders in Atari tasks. Nonetheless, SPR's analysis focuses on return performance and lacks direct evidence of representational collapse.

Addressing these research gaps, we aim to conduct an extensive comparison of ZP targets in both MDPs and POMDPs. Our analysis includes providing direct evidence through the estimation of representational rank.

D.2 Limitations and Conclusion

Limitations. The limitations of our work can be divided into theoretical and empirical aspects. On the theoretical side, although we show a continuous-time analysis of auxiliary learning dynamics with linear models in Thm. 3, we do not provide a convergence analysis for the joint optimization of RL and auxiliary losses and the results may not hold beyond linear assumption. On the empirical side, our experiment scope does not cover more complicated domains that require pixel-based observations.

Conclusion. This work has offered a principled analysis of state and history representation learning in reinforcement learning, bridging the gap between various approaches. Our unified view and analysis of self-predictive learning also inspire a minimalist RL algorithm for learning self-predictive representations. Extensive empirical studies in benchmarks across standard MDPs, distracting MDPs, and sparse-reward POMDPs, validate most of our hypotheses suggested by our theory.

E Experimental Details

E.1 Small Scale Experiments to Illustrate Thm. 3

In this section, we discuss the details of the experiments used to explore the empirical effects of using stop-gradient to detach the ZP target in the self-predictive loss. First, we discuss the details shared between both domains and then discuss domain-specific details.

We learn on data obtained by rolling out 10 trajectories under a fixed, near-optimal policy starting from a random state. Trajectories are followed until termination or until 200 transition have been observed, whichever happens first. The encoder, $\phi \in \mathbb{R}^{k\times 2}$ where k is the number of observed features, is updated using full gradient descent with a small learning rate, $\alpha = 0.01$, for 500 steps. At every 10 steps, the absolute cosine similarity between the 2 columns of ϕ is computed, i.e., $f(x, y) = |x^\top y|/(||x||_2||y||_2)$ and the results are plotted in Fig. 2. The optimal transition model $\theta^* = \begin{bmatrix}\theta_z^{*\top} & \theta_a^{*\top}\end{bmatrix}^\top$ is solved using singular value decomposition and the Moore-Penrose inverse to minimize the linear least-squares objective:

$$\left\| \begin{bmatrix}\phi^\top S & A\end{bmatrix} \begin{bmatrix}\theta_z \\ \theta_a\end{bmatrix} - \widetilde{\phi}^\top S' \right\|_2, \tag{149}$$

where S and S' are matrices with each row corresponding to the sampled states (histories) and next states (histories), respectively, and, similarly, A is a row-wise matrix of the sampled actions. The $\widetilde{\phi}$ is set as ϕ in online target, or $\bar{\phi}$ in detached target and EMA target where the Polyak step size $\tau = 0.005$. To avoid numerical issues, singular values close to zero are discarded according to the default behavior of JAX's (**?**) `jax.numpy.linalg.lstsq` method when using `float32` encoding.

Mountain car (Moore, 1990). We follow the dynamics and parameters used in (Sutton & Barto, 2018, Example 10.1). We encode states using a 10×10 uniform grid of radial basis function (RBF), e.g., $f_i(s) = \exp(-(s-c_i)^\top \Sigma^{-1}(s-c_i))$ for an RBF centered on c_i, and with a width corresponding to 0.15 of the span of the state space. Specifically, Σ is diagonal and normalizes each dimension such that the width of the RBF covers 0.15 in each dimension. As a result, the total number of features $k = 100$. Actions are encoded using one-hot encoding and $|\mathcal{A}| = 3$. The policy used to generate data is an energy pumping policy which always picks actions that apply a force in the direction of the velocity and applies a negative force when the speed is zero.

Load-unload (Meuleau et al., 2013). Load-unload is a POMDP with 7 states arranged in a chain. There are 2 actions which allow the agent to deterministically move left or right along the chain, while attempting to move past the left-most or right-most state results in no movement. There are three possible observations which deterministically correspond to being in the left-most state, the right-most state or in any one of the 5 intermediate states. Observations and actions are encoded using one-hot encodings. The agent's state correspond to the history of observation and actions over a fixed window of size 20 with zero padding for a total of $k = 98$ features ($k = 20 \times 3 + 19 \times 2$). Finally, the policy used to generate trajectories is a stateful policy that repeats the last action with probability 0.8 and always starting with the `move-left` action.

E.2 MDP Experiments in Sec. 5.1 and Sec. 5.2

Standard MuJoCo in Sec. 5.1. This is a popular continuous control benchmark from OpenAI Gym (Brockman et al., 2016). We evaluate on Hopper-v2 (11-dim), Walker2d-v2 (17-dim), HalfCheetah-v2 (17-dim), Ant-v2 (111-dim), and Humanoid-v2 (376-dim), where the numbers in the brackets are observation dimensions.

Distracting MuJoCo in Sec. 5.2. We follow Nikishin et al. (2022) to augment the state space with a distracting dimension in Hopper-v2, Walker2d-v2, HalfCheetah-v2, and Ant-v2. The number of distractors varies from $2^4 = 16$ to $2^8 = 256$. Therefore, the largest observation dimension is $256 + 111 = 367$ in distracting Ant-v2. The distractors follow i.i.d. standard Gaussian $\mathcal{N}(0, I)$.

Our algorithm setup in Sec. 5.1 and Sec. 5.2 largely follows the code of ALM(3) (Ghugare et al., 2022)[16]. The original ALM paper also introduces an ablation of the method, "ALM-no-model", which uses model-free RL (rather than SVG) to update the actor parameters. This ablation is structurally similar to our method, which similarly avoids using a model. However, ALM-no-model still employs a reward model and a latent model for learning representations, although not for updating policy.

Below, we compare ALM and our minimalist ϕ_L implementation. We show ablation results comparing our method and ALM variants in Sec. G.

Differences between our minimalist ϕ_L (with reverse KL and EMA targets) and ALM.

- **Reward model**: we remove reward models from both ALM(3) and ALM-no-model. It should be noted that although ALMs learn reward models, they do not update their encoders through reward prediction loss.
- **Encoder objective**: our state encoder (ϕ) is updated by Eq. 5. Given a probabilistic encoder $\mathbb{P}_\phi(z \mid s)$ and a probabilistic latent model $\mathbb{P}_\theta(z' \mid z, a)$ for MDPs, and the latent state $z_\phi \sim \mathbb{P}_\phi(z \mid s)$, we formulate our encoder objective for a data tuple (s, a, r, s') as follows:

$$\min_\phi \quad (Q_\omega(z_\phi, a) - Q^{\text{tar}}(s, a, s', r))^2 - Q_\omega(z_{\overline{\phi}}, \pi_\nu(z_\phi)) + D_{\text{KL}}(\mathbb{P}_\theta(z' \mid z_\phi, a) \,||\, \mathbb{P}_{\overline{\phi}}(z' \mid s')). \tag{150}$$

  In contrast, **ALM-no-model** employs a more complicated objective to train the state encoder ϕ. It performs a 1-step rollout with the reward model $R_\mu(z, a)$ without a discount factor, and modify the stop-gradients on latent states within the Q-value. Given a data tuple (s, a, s'), the objective is

$$\min_\phi \quad - R_\mu(z_\phi, a) - \mathbb{E}_{z'_{\phi,\theta} \sim \mathbb{P}_\theta(|z_\phi, a)} \left[Q_\omega(z'_{\phi,\theta}, \pi_\nu(z'_{\overline{\phi},\theta})) \right] + D_{\text{KL}}(\mathbb{P}_\theta(z' \mid z_\phi, a) \,||\, \mathbb{P}_{\overline{\phi}}(z' \mid s')), \tag{151}$$

  where they eliminate the mean-squared TD loss and maximize the Q-value through its latent states rather than actions, as done in our objective. **ALM(3)** extends ALM-no-model by implementing a 3-step rollout in the encoder objective.

  To isolate the design of stop-gradients in Q-value and mean-squared TD error, we introduce **ALM(0)** that lies between ALM-no-model and ours with the 0-step objective:

$$\min_\phi \quad -Q_\omega(z_\phi, \pi_\nu(z_{\overline{\phi}})) + D_{\text{KL}}(\mathbb{P}_\theta(z' \mid z_\phi, a) \,||\, \mathbb{P}_{\overline{\phi}}(z' \mid s')). \tag{152}$$

[16] https://github.com/RajGhugare19/alm

Table 3: **Hyperparameters used in Markovian agents in standard and distracting MuJoCo.**

Hyperparameter	Value
Discount factor (γ)	0.99
Warmup steps	5000
Target network update rate (τ)	0.005
Replay buffer size	10^6 for Humanoid-v2 and 10^5 otherwise
Batch size	512
Learning rate	0.0001
Max gradient norm	100
Latent state dimension	50
Exploration stddev. clip	0.3
Exploration stddev. schedule	linear$(1.0, 0.1, 100000)$
Auxiliary loss coefficient (λ)	1.0 for ZP-FKL, ZP-RKL and OP, and 10.0 for ZP-ℓ_2

- **Actor objective**: our algorithm share the same actor objective with ALM-no-model and ALM(0), compared to ALM(3) which uses SVG with a 3-step rollout (Heess et al., 2015) and additional intrinsic rewards (*i.e.* the negative reverse KL divergence term; see Eq. 8 in ALM paper for details).

Implementation details for our minimalist algorithm learning ϕ_L, and learning ϕ_O and ϕ_{Q^*}. We follow the exact implementation of the network architectures in ALM(3). The encoder, actor, and critic are parameterized as 2-layer neural networks with 512 hidden units. The latent transition model (only used in learning ϕ_L) and observation predictor (only used in learning ϕ_O) are parameterized as 2-layer networks with 1024 hidden units. The probabilistic encoder, latent model and decoder output a Gaussian distribution with a diagonal covariance matrix. We apply layer normalization (Ba et al., 2016) after the first layer of the critic network. We use ELU activation (Clevert et al., 2015) and Adam optimizers (Kingma & Ba, 2015) for all networks.

We enumerate the values of our hyperparameters in Table 3. If a hyperparameter is shared with ALM(3), we maintain the same value as that used in ALM(3) (Ghugare et al., 2022, Table 3).

E.3 POMDP Experiments in Sec. 5.3

MiniGrid in Sec. 5.3. This is a widely-used discrete gridworld benchmark from Farama foundation (Chevalier-Boisvert et al., 2018; 2023). In this benchmark, an agent has a first-person view to navigate a 2D gridworld with obstacles (*e.g.*, walls and lava). Some tasks require the agent to pick up keys and open doors to navigate to the goal location. The agent's observations are symbolic (not pixel-based) with a size of $7 \times 7 \times 3$ where 7×7 is the spatial field of view, and the 3 channels encode different semantics. The action space is discrete with 7 options: turn left, turn right, move forward, pick up, drop, toggle, and done. Tasks are goal-oriented; the episode terminates immediately when the agent reaches the goal, or times out after a maximum of T steps. Rewards are designed to encourage fast task completion. A successful episode yields a reward of $1 - 0.9 * H/T \in [0.1, 1.0]$ at the terminal step, where H denotes the total steps. Failed episodes result in a reward of 0.0.

We select 20 tasks in MiniGrid, following the recent work RQL-AIS (Seyedsalehi et al., 2023). All of these tasks require memory in an agent. The tasks are grouped as follows:

- **SimpleCrossing** (4 tasks): SimpleCrossingS9N1, SimpleCrossingS9N2, SimpleCrossingS9N3, SimpleCrossing11N5
- **LavaCrossing** (4 tasks): LavaCrossingS9N1, LavaCrossingS9N2, LavaCrossingS9N3, LavaCrossing11N5
- **Unlock** (2 tasks): Unlock, UnlockPickup
- **DoorKey** (3 tasks): DoorKey-5x5, DoorKey-6x6, DoorKey-8x8
- **KeyCorridor** (3 tasks): KeyCorridorS3R1, KeyCorridorS3R2, KeyCorridorS3R3
- **ObstructedMaze** (2 tasks): ObstructedMaze-1Dl, ObstructedMaze-1Dlh

- **MultiRoom** (2 tasks): MultiRoom-N2-S4, MultiRoom-N4-S5

In each group, we arrange the tasks by increasing level of difficulty. Please refer to the MiniGrid website[17] for detailed descriptions. Note that while RQL-AIS also evaluates the RedBlueDoors tasks, we have omitted them from our selection as they are MDPs.

Implementation details on algorithms. We adopt the RQL-AIS codebase[18] for our implementation of a non-distributed version of R2D2 (Kapturowski et al., 2018). We retain their exact implementation and hyperparameters for R2D2, which includes a recurrent replay buffer with uniform sampling, a 50-step burn-in period, a 10-step rollout, and a stepsize of 5 for multi-step double Q-learning. The only difference is that we replace the periodic hard update of target networks with a soft update, to align with our EMA setting.

We implement our end-to-end approaches based on R2D2. Minimalist ϕ_L introduces a single auxiliary task of ZP; while ϕ_O adds a single auxiliary task of OP. Both use deterministic ℓ_2 loss. We normalize the loss coefficient λ Eq. 5 by the output dimension (*i.e.*, 128 for ZP loss and 147 for OP loss) to balance with Q-learning scalar loss. We tune the normalized λ between $(0.01, 0.03, 0.3, 1.0, 3.0, 10.0, 100.0)$ in SimpleCrossing and LavaCrossing tasks. We find that 0.01 works best for OP and 1.0 best for ZP.

Furthermore, we also implement the phased approaches based on R2D2. Both ϕ_L (RP + ZP) and ϕ_O (RP + OP) freeze the encoder parameters during Q-learning. We introduce the coefficient α multiplied to ZP or OP loss to integrate with RP loss. All three losses use deterministic ℓ_2 objectives. We normalize α to balance reward scalar loss and tune it between $(0.01, 0.1, 0.3, 1.0, 3.0, 10.0)$ in SimpleCrossing and LavaCrossing tasks. We find 1.0 works best for both RP + ZP and RP + OP.

We enumerate the values of our hyperparameters in Table 4. If a hyperparameter is shared with R2D2 implemented by RQL-AIS, we maintain the same value as that used in RQL-AIS paper (Seyedsalehi et al., 2023, Table 3). Our network architecture exactly follows RQL-AIS (see their Appendix F).

Lastly, it is important to highlight the distinction between our implementation of RP + OP and the original RQL-AIS approach (Seyedsalehi et al., 2023). Both approaches aim to learn ϕ_O in a phased manner with the same architecture. The main differences are:

- RQL-AIS employs a pre-trained autoencoder to compress the 147-dimensional observations into 64-dimensional latent representations. Then RQL-AIS trains their agent using latent representations while keeping the autoencoder parameters frozen. In contrast, our RP + OP implementation removes the autoencoder and instead directly predicts raw observations.
- RQL-AIS uses MMD loss for observation prediction, which we show is equivalent to learning EZP condition (see our discussion in Sec. C on AIS). Thus, in RP + OP implementation, we replace the MMD loss with an ℓ_2 loss.
- The loss coefficient α is set to 0.5 in RQL-AIS while $0.1/147$ in our implementation.
- We use soft update on target Q network to align with our other implementations.

Despite these implementation differences, we find that our RP + OP implementation performs *similarly* to RQL-AIS across the 20 tasks.

E.4 Evaluation Metrics

We evaluate the **episode return** by executing the deterministic version of the actor to compute the undiscounted sum of rewards.

We estimate the **rank** of a batch of latent states by calling `torch.linalg.matrix_rank(atol, rtol)` function in PyTorch (Paszke et al., 2019). This function calculates the number of singular values that are greater than $\max(\texttt{atol}, \sigma_1 * \texttt{rtol})$ where σ_1 is the largest singular value. In MDP experiments, the batch has a size of $(512, 50)$ with `atol=1e-2, rtol=1e-2`. In POMDP experiments, the batch has a size of $(256 * 10, 128)$

[17] https://minigrid.farama.org/environments/minigrid/
[18] https://github.com/esalehi1996/POMDP_RL

Table 4: **Hyperparameters used in recurrent agents in MiniGrid.**

Hyperparameter	Value
Discount factor (γ)	0.99
Number of environment steps	$4 * 10^6$
Target network update rate (τ)	0.005
Replay buffer size	full
Batch size	256
Learning rate	0.001
Latent state dimension	128
Epsilon greedy schedule	exponential$(1.0, 0.05, 400000)$
R2D2 sequence length	10
R2D2 burn-in sequence length	50
n-step TD	5
Training frequency	every 10 environment steps
Auxiliary loss coefficient (λ)	$1.0/128$ for ZP and $0.01/147$ for OP
Loss coefficient for phased training (α)	$1.0/128$ for RP + ZP and $1.0/147$ for RP + OP

with `atol=1e-3, rtol=1e-3`, where we reshape the 3D tensor of $(256, 10, 128)$ size into 2D matrix.

Each algorithm variant of the experiments is conducted across 12 individual runs in MDPs and 9 individual runs in POMDPs.

We employ the Rliable library (Agarwal et al., 2021) to compute the IQM and its CI for the aggregated curves (Fig. 6). Essentially, IQM is the 25% trimmed mean over the data on 20 tasks with 9 seeds, *i.e.*, 180 runs.

E.5 Computational Resources

It requires around 1.5 days for us to train our algorithm in a (distracting) MuJoCo task for 1.5M environment steps with 3 runs executed in parallel. The 3 runs share a single A100 GPU and utilize 3 CPU cores.

On the same machine, training cost (in secs) per update for Ant-v2 is as follows: model-free agents take around 0.032s, self-predictive and observation-predictive agents with ℓ_2 objective take around 0.036s (13% more), self-predictive and observation-predictive agents with KL objective take around 0.038s (19% more), ALM(3) agent takes around 0.058s (81% more). The brackets show the percentage increase compared to model-free agents.

It requires around 0.5 days for us to train our algorithm in a MiniGrid task for 4M environment steps with 3 runs executed in parallel. The 3 runs share a single V100 GPU and utilize 3 CPU cores.

F Architecture and Code

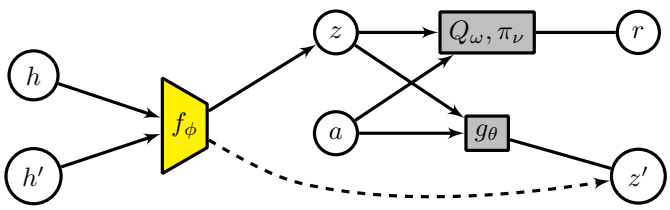

Figure 9: **Architecture of our minimalist ϕ_L algorithm.** The dashed edge indicates the stop-gradient operator; the undirected edges indicate learning from grounded signals of rewards or next latent states.

Besides Algo. 1, we provide a pseudocode written in PyTorch syntax (Paszke et al., 2019) in Algo. 2. Fig. 9 shows our architecture. We also have open-sourced our code at https://github.com/twni2016/self-predictive-rl.

Algorithm 2 Our loss function for learning self-predictive representation in PyTorch syntax

```
def loss(hist, act, next_obs, rew):
    # hist:(B,T,O+A), act:(B,A), next_obs:(B,O), rew:(B,1)
    from torch import cat; from copy import deepcopy

    # Encode histories into latent states
    h_enc = Encoder(hist) #(B,Z)
    next_hist = cat([hist, cat([act, next_obs], dim=-1)], dim=1) #(B,T+1,
    O+A)
    next_h_enc_tar = Encoder_Target(next_hist) #(B,Z)

    # Compute RL loss
    td_tar = rew + gamma * Critic_Target(next_h_enc_tar, Actor(
    next_h_enc_tar))
    critic_loss = ((Critic(h_enc, act)- td_tar.detach())**2).mean()
    actor_loss = -deepcopy(Critic)(h_enc.detach(), Actor(h_enc)).mean()

    # Compute ZP loss
    zp_loss = ((Latent_Model(h_enc, act) - next_h_enc_tar)**2).sum(-1).
    mean()

    return critic_loss + actor_loss + zp_coef * zp_loss
```

G Additional Empirical Results

Ablation studies comparing our minimalist ϕ_L to ALM variants. Fig. 10 shows the comparison between our method and several ALM variants (introduced in Sec. E.2). Both ALM(3) and ALM-no-model have similar encoder objectives (1-step versus 3-step) but differ in actor objective (model-free TD3 versus model-based SVG); their performance is similar except for the Humanoid-v2 tasks, suggesting that the model-based actor update is most useful on higher-dimensional tasks. Comparing ALM-no-model and ALM(0), which only differ in encoder objective (1-step versus 0-step), we see that ALM(0) performs notably worse. This suggests that the use of stop-gradients in Q-value and the omission of mean-squared TD-error in Eq. 152 and Eq. 151 might be problematic, although this issue can be considerably mitigated by the 1-step variant. Finally, our minimalist ϕ_L Eq. 150 performs comparably to ALM-no-model on most tasks and significantly outperforms ALM(0). These results suggest that our method can achieve the benefits of a 1-step rollout without having to unroll a model; however, a method that uses a 3-step rollout can sometimes achieve better results.

Additional results on ZP targets in standard MuJoCo. Fig. 11 shows the performance and the estimated representational rank for the ZP KL divergences (FKL and RKL). Similar to findings in ℓ_2 objective (Fig. 4), we notice significantly lower returns when removing the stop-gradient version (Detached and EMA). Surprisingly, this decreased performance does not seem to be caused by dimensional collapse; on most tasks, the online version of the KL objective does not suffer from dimensional collapse observed for the ℓ_2 objective. These findings suggest that our estimated representational rank may not be correlated with expected returns.

Ablation studies on ZP loss in standard MuJoCo. Fig. 12 shows the ZP losses (ℓ_2 loss, FKL loss, RKL loss) for each ZP objective within our minimalist ϕ_L algorithm. We include results for the online, detached, and EMA targets. As expected, online ZP targets directly minimize ZP losses, thus reaching much lower ZP loss values. However, a lower ZP loss value does not imply higher returns, since the agent needs to balance the RL loss and ZP loss. In future work, we aim to explore strategies to effectively decrease ZP loss without compromising the performance of the stop-gradient variant.

Failed experiments in standard MuJoCo. We did not explore the architecture design and did little hyperparameter tuning on our algorithm. Nevertheless, we observed two failure cases. To match the assumption of Thm. 3 that the gradient w.r.t. the latent transition parameters θ reaches zero, we experimented with higher learning rates $(0.1, 0.01, 0.001)$ for updating the parameters θ in MuJoCo tasks. Yet, we did not observe any performance increase compared to the default learning rate (0.0001). Secondly, inspired by the findings in Fig. 12, we tried a constrained optimization on auxiliary task to adaptively update the loss coefficient using gradient descent-ascent for the stop-gradient version. However, this resulted in a significant performance decline without an explicit decrease in ZP loss values.

Full per-task curves in distracting MuJoCo. Fig. 13 shows all learning curves in distracting MuJoCo.

Full per-task curves in MiniGrid. Fig. 14 shows all learning curves in MiniGrid tasks. Minimalist ϕ_L (ZP) is better than model-free RL (R2D2) in 8 of 20 tasks, and similar in the others except for a single task. On the other hand, ϕ_O (OP) is better than model-free RL (R2D2) in 10 tasks, with the other tasks being identical. Since MiniGrid tasks are deterministic without distraction and the observation is not high-dimensional, ϕ_O (OP) outperforming ϕ_L (ZP) in 7 tasks is expected. The end-to-end ϕ_L (ZP) surpasses its phased counterpart (RP + ZP) in 14 tasks, with the rest tasks being the same. The end-to-end ϕ_O (OP) is better than its phased counterpart (RP + OP) in 7 tasks, but falls short in 4 tasks. These findings underline the efficacy of the end-to-end approach to learning ϕ_L over the phased approach.

Fig. 15 shows all matrix rank curves in MiniGrid tasks. Across all 20 tasks, online ZP targets consistently have the *lowest* matrix rank, aligned with our prediction from Thm. 3. However, while Thm. 3 shows that both detached and EMA targets avoid collapse in *linear* setting, we observe that detached targets severely collapse in 3 tasks, a phenomenon absent with EMA targets. This prompts further theoretical investigation in a *nonlinear* context. As expected, OP consistently achieves the highest rank compared to ZP and R2D2, since OP learns the finest abstraction.

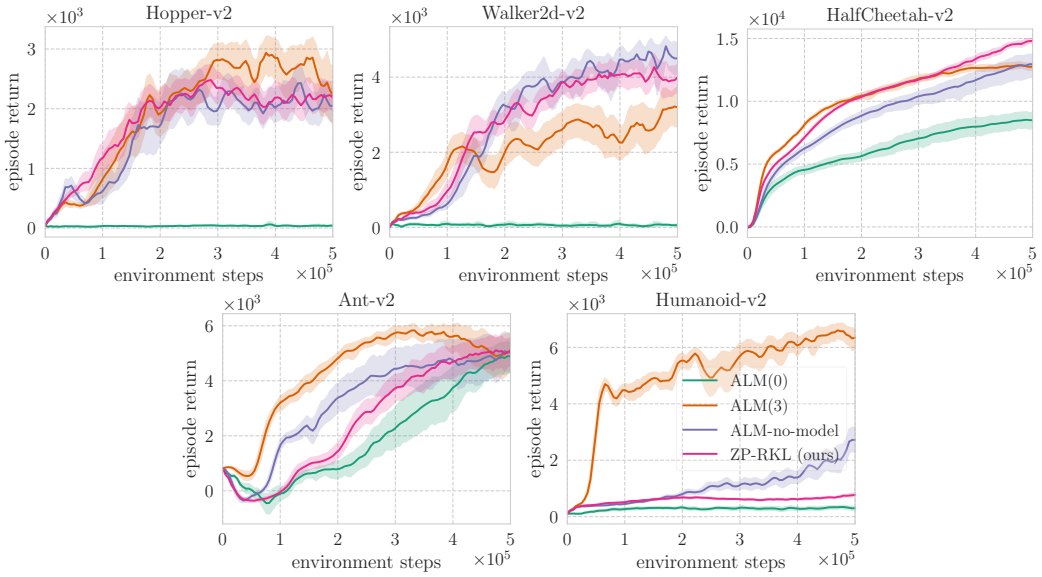

Figure 10: Ablation studies on ALM variants (ALM(3), ALM-no-model, ALM(0)) and our minimalist ϕ_L (ZP-RKL with EMA targets).

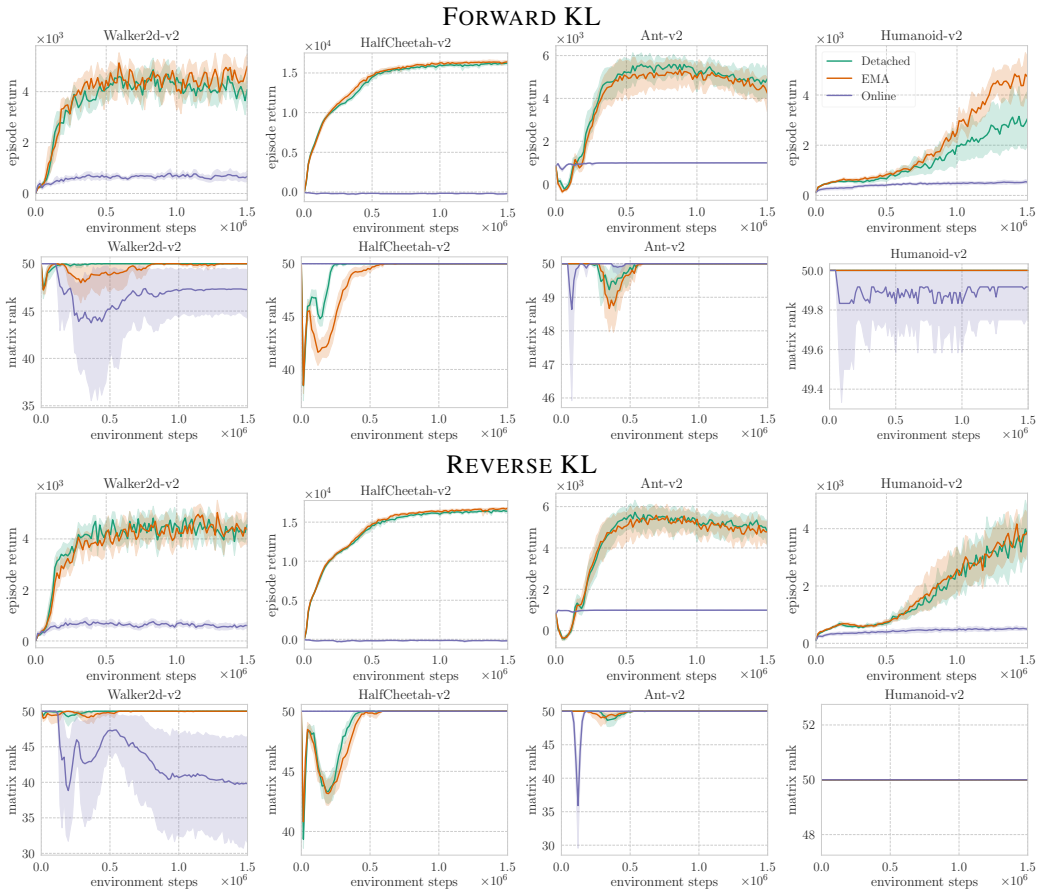

Figure 11: **Representation collapse is less severe with online targets for KL objectives.** On four benchmark tasks, we observe that using the online ZP target results in lower returns. Rows 1 and 2 show results for the forward KL; rows 3 and 4 show result for the reverse KL.

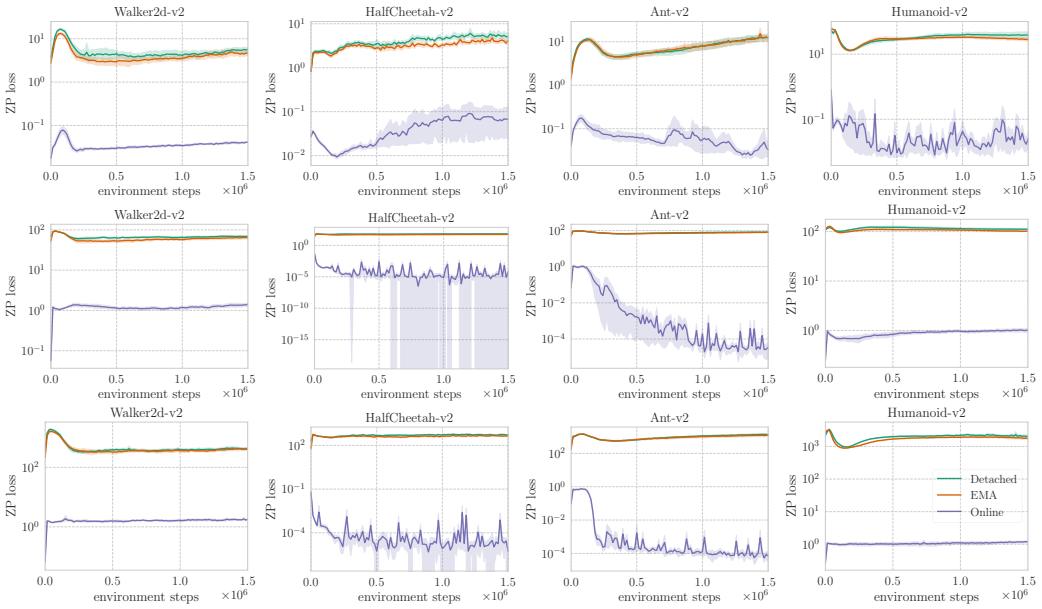

Figure 12: **Online ZP targets reach much smaller values on ZP objectives than stop-gradient ZP targets in standard MuJoCo.** Top row: ZP with ℓ_2 objective; Middle row: ZP with FKL objective; Bottom row: ZP with RKL objective.

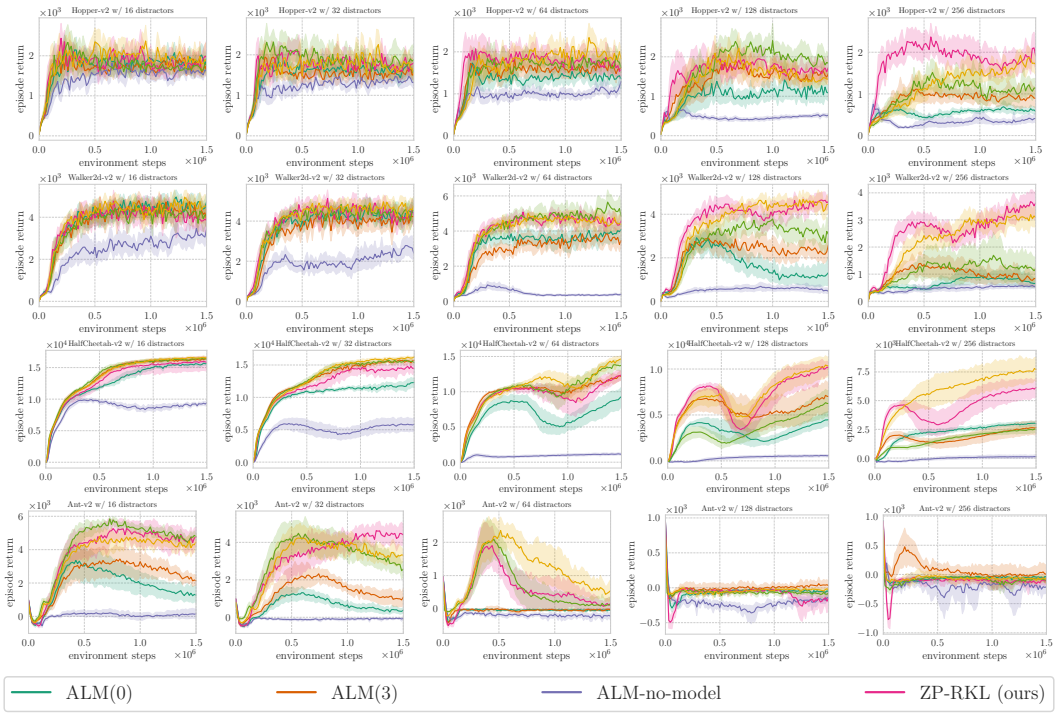

Figure 13: Full learning curves on distracting MuJoCo benchmark.

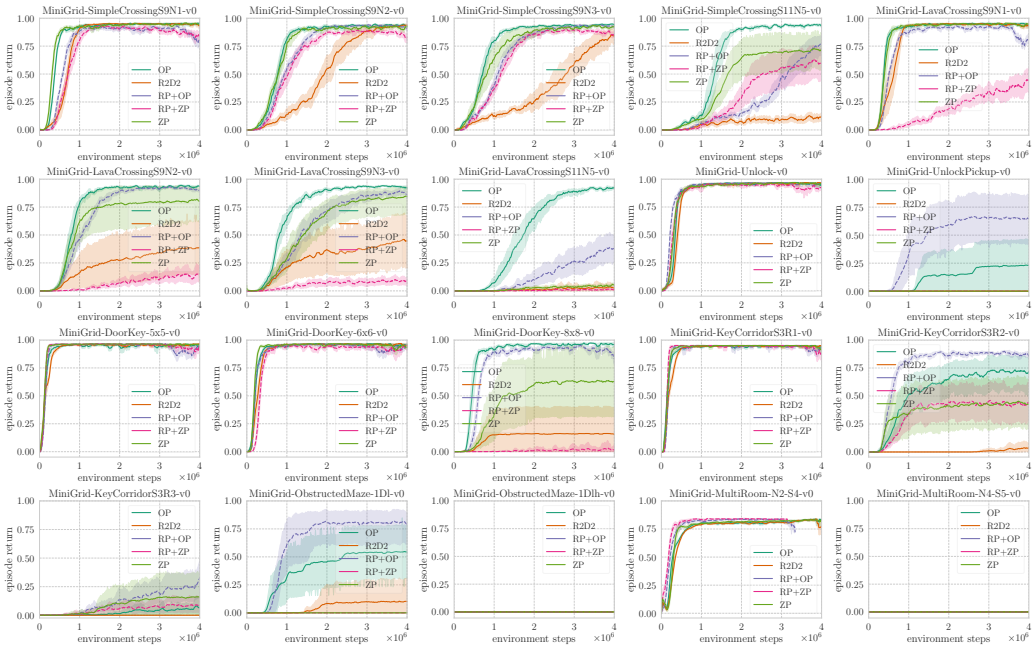

Figure 14: The **episode return** between ϕ_{Q^*} (R2D2), ϕ_L (ZP, RP + ZP), ϕ_O (OP, RP + OP) in 20 MiniGrid tasks over 4M steps, averaged across ≥ 9 seeds. The end-to-end approaches (R2D2, ZP, OP) are shown by **solid** curves, while the phased ones (RP + ZP, RP + OP) are shown by **dashed** curves. The ZP targets use EMA. Tasks sharing the same prefixes (e.g., SimpleCrossing, KeyCorridor) are arranged in order of increasing difficulty.

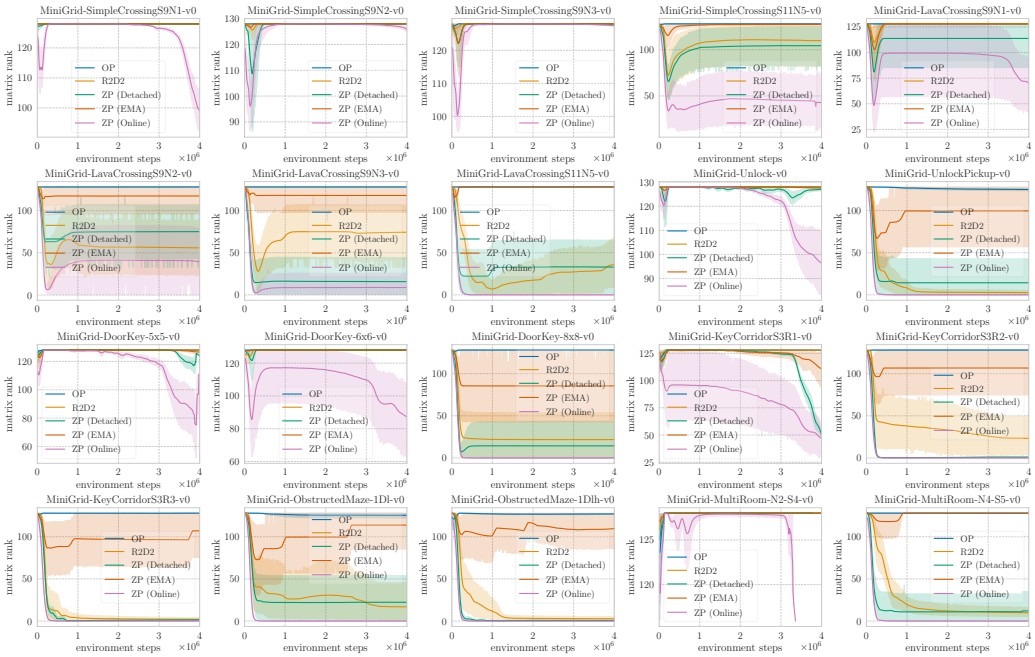

Figure 15: The estimated **matrix rank** between **ZP targets** (online, detached, EMA), R2D2, and OP in 20 MiniGrid tasks over 4M steps, averaged across ≥ 9 seeds. The maximal achievable rank is 128.