Bridging State and History Representations: Understanding Self-Predictive RL

Tianwei Ni[1] **Benjamin Eysenbach**[2] **Erfan Seyedsalehi**[3] **Michel Ma**[1]

Clement Gehring[1] **Aditya Mahajan**[3] **Pierre-Luc Bacon**[1]

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 stop-gradient technique for *learning* self-predictive representations.

1 Introduction

Reinforcement learning holds great potential to 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 [69, 79, 81]. This challenge becomes even more severe in partially observable environments [51] where a history of observations grows over time.

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 [10, 11, 46], history abstraction in POMDPs [6, 47], and sufficient statistics in stochastic control [4, 39, 40, 70]. Traditionally, this compression has been achieved through hand-crafted feature extractors [37, 73] or with the discovery of a set of core tests sufficient for predicting future observations [47, 68]. Modern approaches learn the latent state space using an encoder to automatically filter out irrelevant parts of observations [42, 52, 82]. 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 (the "**what**") in the context of RL, as well as 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 [71]. Next, we examine how to learn such self-predictive condition in RL, a difficult subtask due to the bootstrapping effect [15, 62, 78]. 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 preventing representational collapse in POMDPs. Taken together, we believe that our work may prove helpful for charting a path forward for studying the longstanding challenge of learning representations in MDPs and POMDPs.

[1]Mila, Université de Montréal. [2]Princeton University. [3]Mila, McGill University.

NeurIPS 2023 Workshop: Self-Supervised Learning - Theory and Practice.

2 Background

MDPs and POMDPs. In the context of a POMDP $\mathcal{M}_O = (\mathcal{O}, \mathcal{A}, P, R, \gamma, T)$ [71], 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 $h_t := (h_{t-1}, a_{t-1}, o_t) \in \mathcal{H}_t$, 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 $Q^*(h, a) = \mathbb{E}[r \mid h, a] + \gamma \mathbb{E}_{o' \sim P(\mid h,a)}[\max_{a'} Q^*(h', a')]$ 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 $s_t \in \mathcal{S}$.

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 also known as an "abstraction" [46] or a "latent state" [15]. Such encoders are sometimes shared and simultaneously updated by downstream components (*e.g.* policy, value, reward model, world model) of an RL system [22, 26]. In this paper, we are interested in such a shared encoder, or **the encoder of 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 Appendix A.2.

1. Q^*-irrelevance abstraction. An encoder ϕ_{Q^*} provides a Q^*-irrelevance abstraction [46] 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. In our interpretation, we view the model-irrelevance concept [46] from a self-predictive standpoint. Specifically, a model-irrelevant encoder, denoted as ϕ_L, fulfills two conditions: **expected reward prediction (RP)** and **next latent state z prediction (ZP)**[1], ensuring that the encoder is capable of predicting expected reward and the subsequent latent state distribution.

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

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

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 [17] in MDPs and an information state generator [71] in POMDPs.

3. Observation-predictive (belief) abstraction. This abstraction is implicitly introduced by Subramanian et al. [71], which we denote by ϕ_O, and satisfies three conditions: expected reward prediction RP, **recurrent encoder (Rec)** and **next observation prediction (OP)**[2].

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

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

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

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 [85]. The recurrent encoder (Rec) condition is satisfied for encoders parameterized with feedforward or recurrent neural networks [12, 29], but not Transformers [80]. In this paper, we assume the Rec condition is always satisfied. In POMDPs, ϕ_O is well-known as a belief state generator [34].

[1] RP and ZP are labeled as (P1) and (P2), respectively, in Subramanian et al. [71].

[2] OP and ZM are labeled as (P2a) and (P2b), respectively, in Subramanian et al. [71].

Below we extend the known relations between these abstractions in MDPs [46] to POMDPs.

Theorem 1 (Relationships between common abstractions). *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 the 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 significant novel finding, while postponing the presentation of other propositions and proofs to Sec. A.

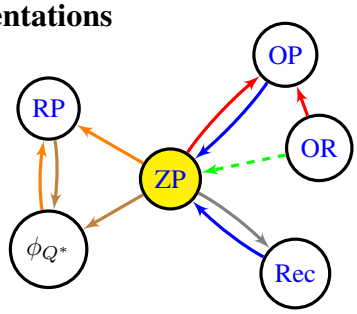

Figure 1: (Better viewed in PDF) **An implication graph** showing the relations between the conditions on history representations. The source nodes of the edges with the same color together imply the target node. The dashed edge means it only applies to MDPs. All the connections are discovered in this 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 (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 (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 $\mathcal{R}_z(\phi(h), a) = \mathbb{E}[R(h, a) \mid h, a], \forall h, a$.*

3.2 Which Representations Do Prior Methods Learn?

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" condition). 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.

Work	PO?	Abstraction	Conditions	ZP loss	ZP target
Model-Free & Classic Model-Based RL	✗	ϕ_{Q^*}	ϕ_{Q^*}	N/A	N/A
DeepMDP [15]	✗	ϕ_L	ϕ_{Q^*} + RP + ZP	W (ℓ_2)	online
SPR [62]	✗	ϕ_L	ϕ_{Q^*} + ZP	cos	EMA
DBC [88]	✗	ϕ_L	ϕ_{Q^*} + RP + ZP + metric	FKL	detached
LSFM [45]	✗	ϕ_L	ϕ_{Q^*} + RP + ZP	SF	detached
Baseline in [79]	✗	ϕ_L	ϕ_{Q^*} + RP + ZP	ℓ_2	detached
EfficientZero [86]	✗	ϕ_L	ϕ_{Q^*} + RP + ZP	cos	detached
TD-MPC [26]	✗	ϕ_L	ϕ_{Q^*} + RP + ZP	ℓ_2	EMA
ALM [16]	✗	ϕ_L	ϕ_{Q^*} + ZP	RKL	EMA
TCRL [89]	✗	ϕ_L	RP + ZP	cos	EMA
OFENet [56]	✗	ϕ_O	ϕ_{Q^*} + OP	N/A	N/A
Recurrent Model-Free RL [27, 36, 53]	✓	ϕ_{Q^*}	ϕ_{Q^*}	N/A	N/A
PBL [18]	✓	ϕ_L	ϕ_{Q^*} + ZP	ℓ_2	detached
AIS [71]	✓	ϕ_L, ϕ_O	RP + ZP or OP	ℓ_2, FKL	detached
Belief-Based Methods [21–25, 44, 83]	✓	ϕ_O	RP + ZP + OR	FKL	online
Causal States [87]	✓	ϕ_O	RP + OP	N/A	N/A

With the unified view of state and history representations, we can categorize prior works by the conditions that their *optimal* encoders in their value functions satisfy. 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 [9, 32, 35, 72, 76], or learn their representations from maximizing returns [54, 67, 77]. Secondly, as shown in Table 1, there is a wealth of prior work on approximating ϕ_L, stemming from different perspectives. These include bisimulation [15], information states [71], variational inference [16], successor features [2, 45], and self-supervised learning [18, 26, 62]. 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), and (3) ZP targets for optimization (including online, detached, EMA, as detailed in Sec. 4). Finally, observation-predictive representations are typically studied in POMDPs, where they are known as belief states [34] and predictive state representations [47].

4 On Learning Self-Predictive Representations in RL

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 to learn ZP, summarized in Table 1's ZP loss column. For simplicity, consider the **deterministic** ℓ_2 **objective** [15, 26, 62, 79, 86][3]:

$$J_\ell(\phi, \theta, \widetilde{\phi}; h, a) := \mathbb{E}_{o' \sim P(|h,a)} \Big[\| g_\theta(f_\phi(h), a) - f_{\widetilde{\phi}}(h') \|_2^2 \Big], \tag{1}$$

where we parameterize an encoder with $f_\phi : \mathcal{H}_t \to \mathcal{Z}$ and a latent transition function with $g_\theta : \mathcal{Z} \times \mathcal{A} \to \mathcal{Z}$. The $\widetilde{\phi}$, 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.

In this section, we aim to justify that the widely used stop-gradient (detached or EMA) ZP targets [16, 62, 88, 89], play an important role in optimization. We find that stop-gradient can avoid representational collapse under some linear assumptions (Thm. 3), while online ZP targets lack these properties.

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 $\mathbb{E}_{h,a}\big[J_\ell(\phi, \theta, \overline{\phi}; h, a)\big]$ without RL loss, and θ relies on ϕ by reaching the stationary point with $\nabla_\theta \mathbb{E}_{h,a}\big[J_\ell(\phi, \theta, \overline{\phi}; h, a)\big] = 0$, then the matrix multiplication $\phi^\top\phi$ will retain its initial value according to our continuous-time analysis.*

Thm. 3 extends the results of [78, 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[4].

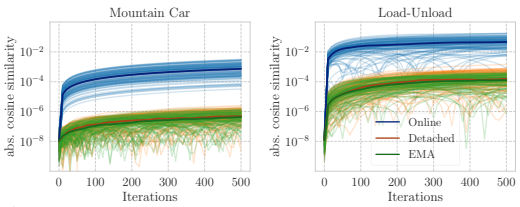

Figure 2: The absolute normalized inner product of the two column vectors in the learned encoder when using online, detached, or EMA ZP target in an MDP *(left)* and a POMDP *(right)*. We plot the results for 100 different seeds, which controls the rollouts used to sample transition and the initialization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

Similar to Tang et al. [78], 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 [50] (MDP) and load-unload [49] (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.

[3]Cosine distance [62] is an ℓ_2 distance on the normalized vector space $\mathcal{Z} = \{ z \in \mathbb{R}^d \mid \|z\|_2 = 1\}$.

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

References

[1] C. Allen, N. Parikh, O. Gottesman, and G. Konidaris. Learning markov state abstractions for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 34:8229–8241, 2021. 10, 24

[2] A. Barreto, W. Dabney, R. Munos, J. J. Hunt, T. Schaul, H. P. van Hasselt, and D. Silver. Successor features for transfer in reinforcement learning. *Advances in neural information processing systems*, 30, 2017. 4, 21, 22

[3] M. G. Bellemare, W. Dabney, and R. Munos. A distributional perspective on reinforcement learning. In *International conference on machine learning*, pages 449–458. PMLR, 2017. 20

[4] T. Bohlin. Information pattern for linear discrete-time models with stochastic coefficients. *IEEE Transactions on Automatic Control*, 15(1):104–106, 1970. 1

[5] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/google/jax`. 25

[6] P. S. Castro, P. Panangaden, and D. Precup. Equivalence relations in fully and partially observable markov decision processes. In *IJCAI*, volume 9, pages 1653–1658, 2009. 1, 12, 16, 23

[7] P. S. Castro, T. Kastner, P. Panangaden, and M. Rowland. Mico: Improved representations via sampling-based state similarity for markov decision processes. *Advances in Neural Information Processing Systems*, 34:30113–30126, 2021. 24

[8] E. Choshen and A. Tamar. Contrabar: Contrastive bayes-adaptive deep rl. *arXiv preprint arXiv:2306.02418*, 2023. 24

[9] K. Chua, R. Calandra, R. McAllister, and S. Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018. 4

[10] P. Dayan. Improving generalization for temporal difference learning: The successor representation. *Neural computation*, 5(4):613–624, 1993. 1, 21

[11] T. Dean and R. Givan. Model minimization in markov decision processes. In *AAAI/IAAI*, pages 106–111, 1997. 1

[12] J. L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990. 2

[13] N. Ferns, P. Panangaden, and D. Precup. Metrics for finite markov decision processes. In *UAI*, volume 4, pages 162–169, 2004. 20

[14] S. Fujimoto, W.-D. Chang, E. J. Smith, S. S. Gu, D. Precup, and D. Meger. For sale: State-action representation learning for deep reinforcement learning. *arXiv preprint arXiv:2306.02451*, 2023. 25

[15] C. Gelada, S. Kumar, J. Buckman, O. Nachum, and M. G. Bellemare. Deepmdp: Learning continuous latent space models for representation learning. In *International Conference on Machine Learning*, pages 2170–2179. PMLR, 2019. 1, 2, 3, 4, 20

[16] R. Ghugare, H. Bharadhwaj, B. Eysenbach, S. Levine, and R. Salakhutdinov. Simplifying model-based rl: Learning representations, latent-space models, and policies with one objective. *arXiv preprint arXiv:2209.08466*, 2022. 3, 4, 21

[17] R. Givan, T. Dean, and M. Greig. Equivalence notions and model minimization in markov decision processes. *Artificial Intelligence*, 147(1-2):163–223, 2003. 2, 11, 12

[18] Z. D. Guo, B. Á. Pires, B. Piot, J. Grill, F. Altché, R. Munos, and M. G. Azar. Bootstrap latent-predictive representations for multitask reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, 2020. 3, 4, 20

[19] Z. D. Guo, S. Thakoor, M. Pîslar, B. A. Pires, F. Altché, C. Tallec, A. Saade, D. Calandriello, J.-B. Grill, Y. Tang, et al. Byol-explore: Exploration by bootstrapped prediction. *arXiv preprint arXiv:2206.08332*, 2022. 20

[20] D. Ha and J. Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 31, 2018. 24

[21] D. Hafner, T. P. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, 2019. 3, 22

[22] D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020. 2

[23] D. Hafner, T. P. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020. 22

[24] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[25] D. Han, K. Doya, and J. Tani. Variational recurrent models for solving partially observable control tasks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020. 3, 22

[26] N. Hansen, X. Wang, and H. Su. Temporal difference learning for model predictive control. *arXiv preprint arXiv:2203.04955*, 2022. 2, 3, 4, 21

[27] M. J. Hausknecht and P. Stone. Deep recurrent Q-learning for partially observable mdps. In *2015 AAAI Fall Symposia, Arlington, Virginia, USA, November 12-14, 2015*, 2015. 3

[28] N. Heess, G. Wayne, D. Silver, T. Lillicrap, T. Erez, and Y. Tassa. Learning continuous control policies by stochastic value gradients. *Advances in neural information processing systems*, 28, 2015. 21

[29] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2

[30] M. Jaderberg, V. Mnih, W. M. Czarnecki, T. Schaul, J. Z. Leibo, D. Silver, and K. Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016. 24

[31] M. R. James, S. Singh, and M. L. Littman. Planning with predictive state representations. In *2004 International Conference on Machine Learning and Applications, 2004. Proceedings.*, pages 304–311. IEEE, 2004. 23

[32] M. Janner, J. Fu, M. Zhang, and S. Levine. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32, 2019. 4

[33] N. Jiang. Notes on state abstractions. http://nanjiang.web.engr.illinois.edu/files/cs598/note4.pdf, 2018. 12

[34] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artif. Intell.*, 1998. 2, 4, 22

[35] L. Kaiser, M. Babaeizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, et al. Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*, 2019. 4

[36] S. Kapturowski, G. Ostrovski, J. Quan, R. Munos, and W. Dabney. Recurrent experience replay in distributed reinforcement learning. In *International conference on learning representations*, 2018. 3

[37] G. Konidaris, S. Osentoski, and P. Thomas. Value function approximation in reinforcement learning using the fourier basis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, pages 380–385, 2011. 1

[38] T. D. Kulkarni, A. Saeedi, S. Gautam, and S. J. Gershman. Deep successor reinforcement learning. *arXiv preprint arXiv:1606.02396*, 2016. 21, 22

[39] P. R. Kumar and P. Varaiya. *Stochastic Systems: Estimation, Identification and Adaptive Control*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1986. ISBN 0-13-846684-X. 1

[40] H. Kwakernaak. *Theory of Self-Adaptive Control Systems*, chapter Admissible Adaptive Control, pages 14–18. Springer, 1965. 1

[41] M. Lange, N. Krystiniak, R. Engelhardt, W. Konen, and L. Wiskott. Comparing auxiliary tasks for learning representations for reinforcement learning, 2023. URL https://openreview.net/forum?id=7Kf5_7-b7q. 23

[42] S. Lange and M. Riedmiller. Deep auto-encoder neural networks in reinforcement learning. In *The 2010 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2010. 1

[43] M. Laskin, A. Srinivas, and P. Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. In *International Conference on Machine Learning*, pages 5639–5650. PMLR, 2020. 24

[44] A. X. Lee, A. Nagabandi, P. Abbeel, and S. Levine. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. 3, 22

[45] L. Lehnert and M. L. Littman. Successor features combine elements of model-free and model-based reinforcement learning. *The Journal of Machine Learning Research*, 21(1):8030–8082, 2020. 3, 4, 21, 22

[46] L. Li, T. J. Walsh, and M. L. Littman. Towards a unified theory of state abstraction for mdps. In *AI&M*, 2006. 1, 2, 3, 10, 11, 12

[47] M. L. Littman, R. S. Sutton, and S. P. Singh. Predictive representations of state. In *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, 2001. 1, 4, 23

[48] B. Mazoure, R. Tachet des Combes, T. L. Doan, P. Bachman, and R. D. Hjelm. Deep reinforcement and infomax learning. *Advances in Neural Information Processing Systems*, 33:3686–3698, 2020. 24

[49] N. Meuleau, K.-E. Kim, L. P. Kaelbling, and A. R. Cassandra. Solving pomdps by searching the space of finite policies. *arXiv preprint arXiv:1301.6720*, 2013. 4, 25

[50] A. W. Moore. Efficient memory-based learning for robot control. Technical report, University of Cambridge, Computer Laboratory, 1990. 4, 25

[51] S. Morad, R. Kortvelesy, M. Bettini, S. Liwicki, and A. Prorok. Popgym: Benchmarking partially observable reinforcement learning. *arXiv preprint arXiv:2303.01859*, 2023. 1

[52] J. Munk, J. Kober, and R. Babuška. Learning state representation for deep actor-critic control. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 4667–4673. IEEE, 2016. 1

[53] T. Ni, B. Eysenbach, and R. Salakhutdinov. Recurrent model-free rl can be a strong baseline for many pomdps. In *International Conference on Machine Learning*, pages 16691–16723. PMLR, 2022. 3, 24

[54] J. Oh, S. Singh, and H. Lee. Value prediction network. *Advances in neural information processing systems*, 30, 2017. 4, 24

[55] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 24

[56] K. Ota, T. Oiki, D. Jha, T. Mariyama, and D. Nikovski. Can increasing input dimensionality improve deep reinforcement learning? In *International conference on machine learning*, pages 7424–7433. PMLR, 2020. 3, 23

[57] G. Patil, A. Mahajan, and D. Precup. On learning history based policies for controlling markov decision processes. *arXiv preprint arXiv:2211.03011*, 2022. 21

[58] J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988. 13

[59] B. Poole, S. Ozair, A. Van Den Oord, A. Alemi, and G. Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, pages 5171–5180. PMLR, 2019. 24

[60] K. Rakelly, A. Gupta, C. Florensa, and S. Levine. Which mutual-information representation learning objectives are sufficient for control? *Advances in Neural Information Processing Systems*, 34:26345–26357, 2021. 24

[61] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020. 20, 24

[62] M. Schwarzer, A. Anand, R. Goel, R. D. Hjelm, A. Courville, and P. Bachman. Data-efficient reinforcement learning with self-predictive representations. *arXiv preprint arXiv:2007.05929*, 2020. 1, 3, 4, 20

[63] D. Sejdinovic, B. Sriperumbudur, A. Gretton, and K. Fukumizu. Equivalence of distance-based and rkhs-based statistics in hypothesis testing. *The annals of statistics*, pages 2263–2291, 2013. 21

[64] E. Seyedsalehi, N. Akbarzadeh, A. Sinha, and A. Mahajan. Approximate information state based convergence analysis of recurrent q-learning. *arXiv preprint arXiv:2306.05991*, 2023. 21

[65] C. R. Shalizi and J. P. Crutchfield. Computational mechanics: Pattern and prediction, structure and simplicity. *Journal of statistical physics*, 104:817–879, 2001. 23

[66] E. Shelhamer, P. Mahmoudieh, M. Argus, and T. Darrell. Loss is its own reward: Self-supervision for reinforcement learning. *arXiv preprint arXiv:1612.07307*, 2016. 24

[67] D. Silver, H. van Hasselt, M. Hessel, T. Schaul, A. Guez, T. Harley, G. Dulac-Arnold, D. P. Reichert, N. C. Rabinowitz, A. Barreto, and T. Degris. The predictron: End-to-end learning and planning. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, 2017. 4

[68] S. P. Singh, M. L. Littman, N. K. Jong, D. Pardoe, and P. Stone. Learning predictive state representations. In *International Conference on Machine Learning (ICML)*, 2003. 1

[69] A. Stone, O. Ramirez, K. Konolige, and R. Jonschkowski. The distracting control suite – a challenging benchmark for reinforcement learning from pixels. *arXiv preprint arXiv:2101.02722*, 2021. 1

[70] C. Striebel. Sufficient statistics in the optimum control of stochastic systems. *Journal of Mathematical Analysis and Applications*, 12(3):576–592, 1965. 1

[71] J. Subramanian, A. Sinha, R. Seraj, and A. Mahajan. Approximate information state for approximate planning and reinforcement learning in partially observed systems. *J. Mach. Learn. Res.*, 23:12–1, 2022. 1, 2, 3, 4, 12, 13, 14, 16, 17, 18, 20, 21

[72] R. S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine learning proceedings 1990*, pages 216–224. Elsevier, 1990. 4

[73] R. S. Sutton. Generalization in reinforcement learning: Successful examples using sparse coarse coding. *Advances in neural information processing systems*, 8, 1995. 1

[74] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018. 25

[75] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999. 10

[76] R. S. Sutton, C. Szepesvári, A. Geramifard, and M. P. Bowling. Dyna-style planning with linear function approximation and prioritized sweeping. *arXiv preprint arXiv:1206.3285*, 2012. 4

[77] A. Tamar, Y. Wu, G. Thomas, S. Levine, and P. Abbeel. Value iteration networks. *Advances in neural information processing systems*, 29, 2016. 4

[78] Y. Tang, Z. D. Guo, P. H. Richemond, B. Á. Pires, Y. Chandak, R. Munos, M. Rowland, M. G. Azar, C. L. Lan, C. Lyle, et al. Understanding self-predictive learning for reinforcement learning. *arXiv preprint arXiv:2212.03319*, 2022. 1, 4, 19

[79] M. Tomar, U. A. Mishra, A. Zhang, and M. E. Taylor. Learning representations for pixel-based control: What matters and why? *arXiv preprint arXiv:2111.07775*, 2021. 1, 3, 4

[80] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[81] T. Wang, X. Bao, I. Clavera, J. Hoang, Y. Wen, E. Langlois, S. Zhang, G. Zhang, P. Abbeel, and J. Ba. Benchmarking model-based reinforcement learning. *arXiv preprint arXiv:1907.02057*, 2019. 1

[82] M. Watter, J. T. Springenberg, J. Boedecker, and M. A. Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 2015. 1, 24

[83] G. Wayne, C.-C. Hung, D. Amos, M. Mirza, A. Ahuja, A. Grabska-Barwinska, J. Rae, P. Mirowski, J. Z. Leibo, A. Santoro, et al. Unsupervised predictive memory in a goal-directed agent. *arXiv preprint arXiv:1803.10760*, 2018. 3

[84] L. Yang, K. Zhang, A. Amice, Y. Li, and R. Tedrake. Discrete approximate information states in partially observable environments. In *2022 American Control Conference (ACC)*, pages 1406–1413. IEEE, 2022. 21

[85] D. Yarats, A. Zhang, I. Kostrikov, B. Amos, J. Pineau, and R. Fergus. Improving sample efficiency in model-free reinforcement learning from images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10674–10681, 2021. 2, 23

[86] W. Ye, S. Liu, T. Kurutach, P. Abbeel, and Y. Gao. Mastering atari games with limited data. *Advances in Neural Information Processing Systems*, 34:25476–25488, 2021. 3, 4, 20

[87] A. Zhang, Z. C. Lipton, L. Pineda, K. Azizzadenesheli, A. Anandkumar, L. Itti, J. Pineau, and T. Furlanello. Learning causal state representations of partially observable environments. *arXiv preprint arXiv:1906.10437*, 2019. 3, 23

[88] A. Zhang, R. McAllister, R. Calandra, Y. Gal, and S. Levine. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*, 2020. 3, 4, 20

[89] Y. Zhao, W. Zhao, R. Boney, J. Kannala, and J. Pajarinen. Simplified temporal consistency reinforcement learning. *arXiv preprint arXiv:2306.09466*, 2023. 3, 4, 21

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

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 [46] 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 [75] 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 [1] in MDPs to POMDPs.

Granularity in relation. In MDPs, it is well-known that state representations form a hierarchical structure [46, 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)). $\phi_O \succeq \phi_L \succeq \phi_{Q^*} \succeq \phi_{\pi^*}$.

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	$\mathbb{E}[r_t \mid h_t, a_t] = \mathbb{E}[r_t \mid \phi(h_t), a_t]$
OR	**O**bservation **R**econstruction	$P(o_t \mid h_t) = P_o(o_t \mid \phi(h_t))$
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	$P(z_{t+1} \mid h_t, a_t) = P_z(z_{t+1} \mid \phi(h_t), a_t)$
Rec	Recurrent Encoder	$z_{t+1} \perp\!\!\!\perp h_t \mid \phi(h_t), a_t, o_{t+1}$
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

Abstract MDP. Given an encoder ϕ, we can construct an abstract MDP [46] $\mathcal{M}_\phi = (\mathcal{Z}, \mathcal{A}, P_z, R_z, \gamma, T)$ for a POMDP $\mathcal{M}_O$. The latent reward R_z and latent transition P_z are then given by: $R_z(z, a) = \int P(h \mid z) R(h, a) dh$, $P_z(z' \mid z, a) = \int P(h \mid z) P(o' \mid h, a) \delta(z' = \phi(h')) dh do'$, 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 [46] (bisimulation relation [17]) 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

$$\mathbb{E}[R(h_i, a) \mid h_i, a] = \mathbb{E}[R(h_j, a) \mid h_j, a], \quad \forall a \in \mathcal{A} \tag{2}$$

$$P(z' \mid h_i, a) = P(z' \mid h_j, a), \quad \forall a \in \mathcal{A}, z' \in \mathcal{Z} \tag{3}$$

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 [17, 46] into POMDPs. It is worth noting that while original concepts assume deterministic rewards or require reward distribution matching for stochastic rewards [6] in Eq. 2, the requirement can indeed be relaxed. As shown by Subramanian et al. [71], 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 1 (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,

$$\mathbb{E}[R(h_i, a) \mid h_i, a] = R_z(\phi_L(h_i), a) = R_z(\phi_L(h_j), a) = \mathbb{E}[R(h_j, a) \mid h_j, a], \tag{4}$$

and by ZP,

$$P(z' \mid h_i, a) = P_z(z' \mid \phi_L(h_i), a) = P_z(z' \mid \phi_L(h_j), a) = P(z' \mid h_j, a) \tag{5}$$

Therefore, ϕ_L implies Φ_L.

Now we want to show Φ_L implies ϕ_L. First, to see RP condition: for any h, a,

$$\mathbb{E}[R(h, a) \mid h, a] = \mathbb{E}[R(h, a) \mid h, a] \int_{\mathcal{H}} P(\overline{h} \mid \Phi_L(h))d\overline{h} \tag{6}$$

$$= \int_{\mathcal{H}} P(\overline{h} \mid \Phi_L(h))\mathbb{E}[R(h, a) \mid h, a]d\overline{h} \tag{7}$$

$$\overset{(A)}{=} \int_{\mathcal{H}} P(\overline{h} \mid \Phi_L(h))\mathbb{E}\left[R(\overline{h}, a) \mid \overline{h}, a\right]d\overline{h} := R_z(\Phi_L(h), a) \tag{8}$$

where

$$P(\overline{h} \mid \Phi_L(h)) = \begin{cases} \frac{1}{c} & \text{if } \Phi_L(\overline{h}) = \Phi_L(h) \\ 0 & \text{else} \end{cases} \tag{9}$$

the normalizing constant $c \in [0^+, \infty)$ is the measure of the inverse image $\Phi_L^{-1}(\Phi_L(h))$.

The step (A) follows that for any $\overline{h}$ such that $P(\overline{h} \mid \Phi_L(h)) > 0$ (*i.e.*, $\Phi_L(\overline{h}) = \Phi_L(h)$), by the definition of Φ_L, we have $\mathbb{E}\left[R(\overline{h}, a) \mid \overline{h}, a\right] = \mathbb{E}[R(h, a) \mid h, a]$. The final equation follows that we can construct a latent reward function with $\Phi_L(h)$ and a as inputs, as $\overline{h}$ is integrated.

Similar to the proof of showing RP, we can show ZP: for any h, a,

$$P(z' \mid h, a) = P(z' \mid h, a) \int_{\mathcal{H}} P(\overline{h} \mid \Phi_L(h))d\overline{h} \tag{10}$$

$$= \int_{\mathcal{H}} P(\overline{h} \mid \Phi_L(h))P(z' \mid h, a)d\overline{h} \tag{11}$$

$$\overset{(B)}{=} \int_{\mathcal{H}} P(\overline{h} \mid \Phi_L(h))P(z' \mid \overline{h}, a)d\overline{h} := P_z(z' \mid \Phi_L(h), a) \tag{12}$$

where the step (B) follows the definition of Φ_L that $P(z' \mid h, a) = P(z' \mid \overline{h}, a)$. □

Belief abstraction Φ_O **(weak belief bisimulation relation [6]).** 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

$$\mathbb{E}[R(h_i, a) \mid h_i, a] = \mathbb{E}[R(h_j, a) \mid h_j, a], \quad \forall a \in \mathcal{A} \tag{13}$$

$$P(o' \mid h_i, a) = P(o' \mid h_j, a), \quad \forall a \in \mathcal{A}, o' \in \mathcal{O} \tag{14}$$

This concept is known as a naive abstraction in MDPs [33] and weak belief bisimulation relation in POMDPs [6]. Similarly, prior concepts assume deterministic reward or distribution matching for stochastic rewards, while we relax it to expected reward matching.

Proposition 2 (Φ_O **is equivalent to** ϕ_O**).**

Proof. The proof is almost the same as the proof of Prop. 1 by replacing z' with o'. □

A.3 Propositions and Proofs

With the additional background in Appendix A.2, we show the complete implication graph in Fig. 3 built on Fig. 1.

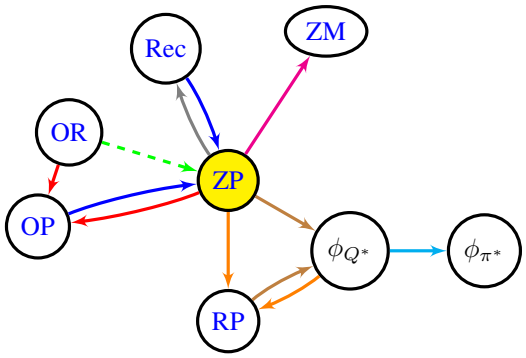

Figure 3: **The complete implication graph** showing the relations between the conditions on history representations. The source nodes of the edges with the same color together imply the target node. The dashed edge means it only applies to MDPs. As a quick reminder, RP: expected reward prediction, OP: 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 (measurable) 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 (measurable) sets,

$$P(f(X) \in A, g(Y) \in B) = P(X \in f^{-1}(A), Y \in g^{-1}(B)) \tag{15}$$

$$\overset{X \perp\!\!\!\perp Y}{=} P(X \in f^{-1}(A))P(Y \in g^{-1}(B)) = P(f(X) \in A)P(g(Y) \in B) \tag{16}$$

□

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.

$$P(Y, f(Z) \mid X, Z) = P(f(Z) \mid X, Z)P(Y \mid X, Z, f(Z)) \tag{17}$$

$$= P(f(Z) \mid Z)P(Y \mid Z) = P(f(Z) \mid Z)P(Y \mid Z, f(Z)) = P(Y, f(Z) \mid Z) \tag{18}$$

□

Proposition 3 (ZP implies both ZM and Rec.).

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

*Proof of **Proposition 3 (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. □

*Proof of **Proposition 3 (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. 4) 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 [58].

Then we have Rec ($z' \perp\!\!\!\perp h' \mid \phi(h), a, o'$) by Lemma 2, as (a, o') also appears in the condition. □

Proposition 4 (OP and Rec imply ZP [71].).

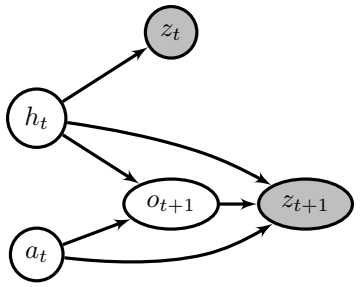

Figure 4: **The graphical model of the interaction between history encoder and the environment.**

*Proof of **Proposition 4 and Theorem 4** ($\phi_O \succeq \phi_L$).* We directly follow the proof in [71, Proposition 4]. Let ϕ satisfy OP and Rec, then we will have ZP:

$$P(z' \mid h, a) = \int P(z', o' \mid h, a) do' = \int P(z' \mid h') P(o' \mid h, a) do' \tag{19}$$

$$\overset{(Rec, OP)}{=} \int P_m(z' \mid \phi(h), a, o') P(o' \mid \phi(h), a) do' \tag{20}$$

$$= \int P(z', o' \mid \phi(h), a) do' = P_z(z' \mid \phi(h), a) \tag{21}$$

□

*Proof of **Theorem 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 5 (OR and ZP imply OP).

Proof. Recall OR is defined as $\delta(o = o) = P(o \mid h) = P(o \mid \phi(h))$[5]. Consider given h, a, for any o',

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

$$\overset{(ZP, OR)}{=} P(\phi(h') \mid \phi(h), a) P(o' \mid \phi(h')) \tag{23}$$

$$= P(\phi(h') \mid \phi(h), a) P(o' \mid \phi(h'), \phi(h), a) \tag{24}$$

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

where Ln. 24 follows that $o' \perp\!\!\!\perp h' \mid \phi(h')$ 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 6 (In MDPs, OR implies ZP and OP).

Proof. Assume ϕ satifies OR in MDPs, *i.e.* $P(s \mid \phi(s)) = \delta(s = s)$. We want to show that $z', s \perp\!\!\!\perp s \mid \phi(s), a$ which implies ZP by Lemma 1. In fact,

$$P(z', s \mid s, \phi(s), a) = P(z', s \mid s, a) = P(z' \mid s, a) \delta(s = s) \tag{26}$$

$$P(z', s \mid \phi(s), a) = P(s \mid \phi(s), a) P(z' \mid s, \phi(s), a) \tag{27}$$

$$\overset{OR}{=} \delta(s = s) P(z' \mid s, a) \tag{28}$$

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

[5] If there is no confusion, we may omit the subscripts of P_z and P_o for notational simplicity.

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. 3.

Proposition 7 (ZP is equivalent to multi-step ZP). *For $k \in \mathbb{N}^+$, define k-step ZP as*

$$P(z_{t+k} \mid h_t, a_{t:t+k-1}) = P(z_{t+k} \mid \phi(h_t), a_{t:t+k-1}), \quad \forall h, a, z \tag{29}$$

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

$$P(z_{t+k} \mid h_t, a_{t:t+k-1}) = \int P(z_{t+1:t+k}, o_{t+1:t+k} \mid h_t, a_{t:t+k-1}) do_{t+1:t+k} dz_{t+1:t+k-1} \tag{30}$$

$$= \int \prod_{i=1}^{k} \delta(z_{t+i} = \phi(h_{t+i})) P(o_{t+i} \mid h_{t+i-1}, a_{t+i-1}) do_{t+1:t+k} dz_{t+1:t+k-1} \tag{31}$$

$$= \int \left(\int \delta(z_{t+k} = \phi(h_{t+k})) P(o_{t+k} \mid h_{t+k-1}, a_{t+k-1}) do_{t+k} \right) \tag{32}$$

$$\prod_{i=1}^{k-1} \delta(z_{t+i} = \phi(h_{t+i})) P(o_{t+i} \mid h_{t+i-1}, a_{t+i-1}) do_{t+1:t+k-1} dz_{t+1:t+k-1} \tag{33}$$

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

$$\overset{ZP}{=} \int P(z_{t+k} \mid \phi(h_{t+k-1}), a_{t+k-1}) \prod_{i=1}^{k-1} \delta(z_{t+i} = \phi(h_{t+i})) P(o_{t+i} \mid h_{t+i-1}, a_{t+i-1}) do_{t+1:t+k-1} dz_{t+1:t+k-1} \tag{35}$$

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

$$= \dots \tag{37}$$

$$= \int \prod_{i=2}^{k} P(z_{t+i} \mid z_{t+i-1}, a_{t+i-1}) P(z_{t+1} \mid h_t, a_t) dz_{t+1:t+k-1} \tag{38}$$

$$\overset{ZP}{=} \int \prod_{i=2}^{k} P(z_{t+i} \mid z_{t+i-1}, a_{t+i-1}) P(z_{t+1} \mid \phi(h_t), a_t) dz_{t+1:t+k-1} \tag{39}$$

$$\overset{ZM}{=} \int P(z_{t+1:t+k} \mid \phi(h_t), a_{t:t+i-1}) dz_{t+1:t+k-1} \tag{40}$$

$$= P(z_{t+k} \mid \phi(h_t), a_{t:t+i-1}) \tag{41}$$

□

Proposition 8 (ZP and RP imply multi-step RP). *For $k \in \mathbb{N}^+$, define k-step RP as*

$$\mathbb{E}[r_{t+k} \mid h_t, a_{t:t+k}] = \mathbb{E}[r_{t+k} \mid \phi(h_t), a_{t:t+k}], \quad \forall h, a \tag{42}$$

Proof.

$$\mathbb{E}[r_{t+k} \mid h_t, a_{t:t+k}] = \int P(o_{t+1:t+k} \mid h_t, a_{t:t+k-1}) \mathbb{E}[r_{t+k} \mid h_{t+k}, a_{t+k}] do_{t+1:t+k} \tag{43}$$

$$\overset{RP}{=} \int P(o_{t+1:t+k} \mid h_t, a_{t:t+k-1}) R_z(\phi(h_{t+k}), a_{t+k}) do_{t+1:t+k} \tag{44}$$

$$= \int P(o_{t+1:t+k} \mid h_t, a_{t:t+k-1}) \delta(z_{t+k} = \phi(h_{t+k})) R_z(z_{t+k}, a_{t+k}) do_{t+1:t+k} dz_{t+k} \tag{45}$$

$$= \int \left(\int P(o_{t+1:t+k} \mid h_t, a_{t:t+k-1}) \delta(z_{t+k} = \phi(h_{t+k})) do_{t+1:t+k} \right) R_z(z_{t+k}, a_{t+k}) dz_{t+k} \tag{46}$$

$$= \int P(z_{t+k} \mid h_t, a_{t:t+k-1}) R_z(z_{t+k}, a_{t+k}) dz_{t+k} \tag{47}$$

$$\overset{k\text{-step ZP}}{=} \int P(z_{t+k} \mid \phi(h_t), a_{t:t+k-1}) R_z(z_{t+k}, a_{t+k}) dz_{t+k} \tag{48}$$

$$= \mathbb{E}[r_{t+k} \mid \phi(h_t), a_{t:t+k}] \tag{49}$$

where k-step ZP is implied by ZP by Prop. 7. □

Proposition 9 (OP implies multi-step OP in MDPs, but not POMDPs). *For $k \in \mathbb{N}^+$, define k-step OP as*

$$P(o_{t+k} \mid h_t, a_{t:t+k-1}) = P(o_{t+k} \mid \phi(h_t), a_{t:t+k-1}), \quad \forall h, a, o \tag{50}$$

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

$$P(s_{t+k} \mid s_t, a_{t:t+k-1}) = \int P(s_{t+1:t+k} \mid s_t, a_{t:t+k-1}) ds_{t+1:t+k-1} \tag{51}$$

$$\overset{\text{MDPs}}{=} \int P(s_{t+1} \mid s_t, a_t) \prod_{i=2}^{k} P(s_{t+i} \mid s_{t+i-1}, a_{t+i-1}) ds_{t+1:t+k-1} \tag{52}$$

$$\overset{OP}{=} \int P(s_{t+1} \mid \phi(s_t), a_t) \prod_{i=2}^{k} P(s_{t+i} \mid s_{t+i-1}, a_{t+i-1}) ds_{t+1:t+k-1} \tag{53}$$

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

$$= P(s_{t+k} \mid \phi(s_t), a_{t:t+k-1}) \tag{55}$$

However, in POMDPs, OP does not imply multi-step OP. This can be shown by a counterexample in Castro et al. [6, 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 [71, Theorem 5 and Theorem 25] for finite-horizon and infinite-horizon POMDPs, respectively. For the approximate version, please see [71, 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 **Theorem 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:

$$\mathcal{Q}(\phi(h_t), a_t) = \begin{cases} \mathbb{E}[R(h_t, a_t) \mid h_t, a_t] & t = T \\ \mathbb{E}[R(h_t, a_t) \mid h_t, a_t] + \gamma \mathbb{E}_{o_{t+1} \sim P(\mid h_t, a_t)} \left[\max_{a_{t+1}} \mathcal{Q}(\phi(h_{t+1}), a_{t+1}) \right] & \text{else} \end{cases} \tag{56}$$

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:

$$Q^*(h_t, a_t) = \begin{cases} \mathbb{E}[R(h_t, a_t) \mid h_t, a_t] & t = T \\ \mathbb{E}[R(h_t, a_t) \mid h_t, a_t] + \gamma \mathbb{E}_{o_{t+1} \sim P(\mid h_t, a_t)} \left[\max_{a_{t+1}} Q^*(h_{t+1}, a_{t+1}) \right] & \text{else} \end{cases} \tag{57}$$

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[6]:

$$\mathcal{R}_z(z_t, a_t) := \begin{cases} \mathcal{Q}(z_t, a_t) & t = T \\ \mathcal{Q}(z_t, a_t) - \gamma \mathbb{E}_{z_{t+1} \sim P(|z_t, a_t)} \left[\max_{a_{t+1}} \mathcal{Q}(z_{t+1}, a_{t+1}) \right] & \text{else} \end{cases} \tag{58}$$

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

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

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

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

$$\mathbb{E}_{o_{t+1} \sim P(|h_t, a_t)} \left[\max_{a_{t+1}} Q^*(h_{t+1}, a_{t+1}) \right] = \mathbb{E}_{z_{t+1} \sim P(|\phi(h_t), a_t)} \left[\max_{a_{t+1}} \mathcal{Q}(z_{t+1}, a_{t+1}) \right], \quad \forall h_t, a_t, t \tag{60}$$

Proof for this:

$$\begin{aligned} \text{LHS} &\overset{\phi_{Q^*}}{=} \mathbb{E}_{o_{t+1} \sim P(|h_t, a_t)} \left[\max_{a_{t+1}} \mathcal{Q}(\phi(h_{t+1}), a_{t+1}) \right] && (61) \\ &= \int \left(\int P(o_{t+1} \mid h_t, a_t) \delta(z_{t+1} = \phi(h_{t+1})) do_{t+1} \right) \max_{a_{t+1}} \mathcal{Q}(z_{t+1}, a_{t+1}) dz_{t+1} && (62) \\ &= \int P(z_{t+1} \mid h_t, a_t) \max_{a_{t+1}} \mathcal{Q}(z_{t+1}, a_{t+1}) dz_{t+1} && (63) \\ &\overset{ZP}{=} \int P(z_{t+1} \mid \phi(h_t), a_t) \max_{a_{t+1}} \mathcal{Q}(z_{t+1}, a_{t+1}) dz_{t+1} = \text{RHS} && (64) \end{aligned}$$

□

Lemma 3 (Integral probability metric [71]). *Given by a function class $\mathcal{F}$, integral probability metric (IPM) between two distributions $\mathbb{P}, \mathbb{Q} \in \Delta(\mathcal{Z})$ is*

$$\mathcal{D}_{\mathcal{F}}(\mathbb{P}, \mathbb{Q}) = \sup_{f \in \mathcal{F}} |\mathbb{E}_{x \sim \mathbb{P}}[f(x)] - \mathbb{E}_{y \sim \mathbb{Q}}[f(y)]| \tag{65}$$

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

$$|\mathbb{E}_{x \sim \mathbb{P}}[g(x)] - \mathbb{E}_{y \sim \mathbb{Q}}[g(y)]| \leq \rho_{\mathcal{F}}(g) \mathcal{D}_{\mathcal{F}}(\mathbb{P}, \mathbb{Q}) \tag{66}$$

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:

$$\mathcal{D}_{\mathcal{F}_{\text{TV}}}(\mathbb{P}, \mathbb{Q}) \leq \sqrt{2 D_{\text{KL}}(\mathbb{P} \,||\, \mathbb{Q})}. \tag{67}$$

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

Theorem 5 (Approximate version of Theorem 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 parametrized by $\mathcal{Q}(\phi(h), a)$ to **approximate** $Q^*(h, a)$, namely: $\forall t, h_t, a_t$,*

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

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

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

$$\mathcal{R}_z(z_t, a_t) := \begin{cases} \mathcal{Q}(z_t, a_t) & t = T \\ \mathcal{Q}(z_t, a_t) - \gamma \mathbb{E}_{z_{t+1} \sim P_z(|z_t, a_t)} \left[\max_{a_{t+1}} \mathcal{Q}(z_{t+1}, a_{t+1}) \right] & \text{else} \end{cases} \tag{68}$$

such that

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

$$\textit{where} \quad \epsilon_t = \begin{cases} \alpha_T & t = T \\ \alpha_t + \gamma(\alpha_{t+1} + \rho_{\mathcal{F}}(\mathcal{V}_{t+1})\delta_t) & \text{else} \end{cases} \tag{69}$$

$$\mathcal{V}(z_t) = \max_{a_t} \mathcal{Q}(z_t, a_t) \tag{70}$$

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:

$$\mathbb{E}[R(h_t, a_t) \mid h_t, a_t] = Q^*(h_t, a_t) - \gamma \mathbb{E}_{o_{t+1} \sim P(|h_t, a_t)} \left[\max_{a_{t+1}} Q^*(h_{t+1}, a_{t+1}) \right] \tag{71}$$

$$\mathcal{R}_z(z_t, a_t) = \mathcal{Q}(z_t, a_t) - \gamma \mathbb{E}_{z_{t+1} \sim P(|z_t, a_t)} \left[\max_{a_{t+1}} \mathcal{Q}(z_{t+1}, a_{t+1}) \right] \tag{72}$$

Therefore, the reward gap is upper bounded:

$$|\mathbb{E}[R(h_t, a_t) \mid h_t, a_t] - \mathcal{R}_z(\phi(h_t), a_t)| \tag{73}$$

$$\le |Q^*(h_t, a_t) - \mathcal{Q}(\phi(h_t), a_t)| \tag{74}$$

$$+ \gamma \left| \mathbb{E}_{o_{t+1} \sim P(|h_t, a_t)} \left[\max_{a_{t+1}} Q^*(h_{t+1}, a_{t+1}) \right] - \mathbb{E}_{z_{t+1} \sim P(|\phi(h_t), a_t)} \left[\max_{a_{t+1}} \mathcal{Q}(z_{t+1}, a_{t+1}) \right] \right| \tag{75}$$

$$\le \alpha_t + \gamma \left| \mathbb{E}_{o_{t+1} \sim P(|h_t, a_t)} \left[\max_{a_{t+1}} Q^*(h_{t+1}, a_{t+1}) - \max_{a_{t+1}} \mathcal{Q}(\phi(h_{t+1}), a_{t+1}) \right] \right| \tag{76}$$

$$+ \gamma \left| \mathbb{E}_{o_{t+1} \sim P(|h_t, a_t)} \left[\max_{a_{t+1}} \mathcal{Q}(\phi(h_{t+1}), a_{t+1}) \right] - \mathbb{E}_{z_{t+1} \sim P(|\phi(h_t), a_t)} \left[\max_{a_{t+1}} \mathcal{Q}(z_{t+1}, a_{t+1}) \right] \right| \tag{77}$$

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

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

$$\le \alpha_t + \gamma \alpha_{t+1} + \gamma \left| \mathbb{E}_{z_{t+1} \sim P(|h_t, a_t)}[\mathcal{V}(z_{t+1})] - \mathbb{E}_{z_{t+1} \sim P(|\phi(h_t), a_t)}[\mathcal{V}(z_{t+1})] \right| \tag{80}$$

$$\le \alpha_t + \gamma(\alpha_{t+1} + \rho_{\mathcal{F}}(\mathcal{V}_{t+1})\delta_t) \tag{81}$$

where Eq. 74 is by triangle inequality, Eq. 76 is by triangle inequality and approx. ϕ_{Q^*}, Eq. 78 is by the maximum-absolute-difference inequality $|\max f(x) - \max g(x)| \le \max |f(x) - g(x)|$, Eq. 80 is by approx. ϕ_{Q^*}, and Ln. 81 is by the property of IPM (Eq. 66 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. Further, 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 [71, Lemma 44], if $\gamma L_p < 1$,

$$\rho_{\mathcal{F}}(\mathcal{V}_{t+1}) = \|\mathcal{V}\|_L \le \frac{L_r}{1 - \gamma L_p}, \quad \forall t \tag{82}$$

Thus, the reward difference bound can be rewritten as

$$\epsilon \le (1 + \gamma)\alpha + \frac{\gamma L_r \delta}{1 - \gamma L_p} \tag{83}$$

B Optimization in Self-Predictive RL

*Proof of **Theorem 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.* $h_{t:-k} = \text{vec}(a_{t-k}, o_{t-k+1}, \dots, a_{t-1}, o_t) \in \mathbb{R}^x$,[7] where $x = k(|\mathcal{O}|+|\mathcal{A}|)$. We assume a linear encoder that maps history $h_t \in \mathcal{H}_t$ into z_t:

$$z_t = f_\phi(h_t) := \phi^\top h_{t:-k} \in \mathbb{R}^d \tag{84}$$

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

$$z_{t+1} = g_\theta(z_t, a_t) := \theta_z^\top z_t + \theta_a^\top a_t \in \mathbb{R}^d \tag{85}$$

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:

$$\dot{\phi} = -\mathbb{E}_{h_t,a_t}\left[\nabla_\phi J_\ell(\phi, \theta, \overline{\phi}; h_t, a_t)\right] \tag{86}$$

$$= -\mathbb{E}_{h_t,a_t,o_{t+1}}\left[\nabla_\phi \|\theta_z^\top \phi^\top h_{t:-k} + \theta_a^\top a_t - \overline{\phi}^\top h_{t+1:-k}\|_2^2\right] \tag{87}$$

$$= -\mathbb{E}_{h_t,a_t}\left[(\theta_z^\top \phi^\top h_{t:-k} + \theta_a^\top a_t - \mathbb{E}_{o_{t+1}}\left[\overline{\phi}^\top h_{t+1:-k}\right])^\top \nabla_\phi \theta_z^\top \phi^\top h_{t:-k}\right] \tag{88}$$

$$= -\mathbb{E}_{h_t,a_t}\left[h_{t:-k}(\theta_z^\top \phi^\top h_{t:-k} + \theta_a^\top a_t - \mathbb{E}_{o_{t+1}}\left[\overline{\phi}^\top h_{t+1:-k}\right])^\top\right]\theta_z^\top \tag{89}$$

The gradient of the loss w.r.t. θ_z:

$$\nabla_{\theta_z}\mathbb{E}_{h_t,a_t}\left[J_\ell(\phi, \theta, \overline{\phi}; h_t, a_t)\right] \tag{90}$$

$$= \mathbb{E}_{h_t,a_t}\left[(\theta_z^\top \phi^\top h_{t:-k} + \theta_a^\top a_t - \mathbb{E}_{o_{t+1}}\left[\overline{\phi}^\top h_{t+1:-k}\right])^\top \nabla_{\theta_z}(\theta_z^\top \phi^\top h_{t:-k} + \theta_a^\top a_t)\right] \tag{91}$$

$$= \phi^\top \mathbb{E}_{h_t,a_t}\left[h_{t:-k}(\theta_z^\top \phi^\top h_{t:-k} + \theta_a^\top a_t - \mathbb{E}_{o_{t+1}}\left[\overline{\phi}^\top h_{t+1:-k}\right])^\top\right] \in \mathbb{R}^{d \times d} \tag{92}$$

Therefore, we have

$$\phi^\top \dot{\phi} = -\nabla_{\theta_z}\mathbb{E}_{h_t,a_t}\left[J_\ell(\phi, \theta, \overline{\phi}; h_t, a_t)\right]\theta_z^\top \tag{93}$$

Following the practice in [78], 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

$$\frac{d(\phi^\top \phi)}{dt} = \dot{\phi}^\top \phi + \phi^\top \dot{\phi} = \dot{\phi}^\top \phi + (\dot{\phi}^\top \phi)^\top = 0 \tag{94}$$

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

C Prior Works on State and History Representation Learning

In this section, we provide a concise overview of previous works that learn or approximate self-predictive or model-based representations. 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. 8 (or ϕ_{Q^*} + ZP by Thm. 2), and multi-step ZP can be implied by ZP by Prop. 7.

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

C.1 Self-Predictive Representations

DeepMDP [15]: 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 encoder, transition and latent transition, resulting in dirac distributions $\mathbb{P}_\phi(z' \mid s')$ and $\mathbb{P}_{\phi,\theta}(z' \mid s, a)$. 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 C51 agent [3], outperforming C51 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 [62]: 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:

$$P(z_{t+1:t+k} \mid s_t, a_{t:t+k-1}) = P(z_{t+1:t+k} \mid \phi(s_t), a_{t:t+k-1}) \tag{95}$$

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 [88]: 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 [13] 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 [18]: 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

$$g(\theta(\phi(h), a)) = g(f(o')) = \phi(h') \tag{96}$$

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 [19] removes reverse prediction.

EfficientZero [86]: ϕ_{Q^*} + RP + multi-step ZP with detached cos. EfficientZero improves MuZero [61] 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 [62], they use 5-step cos objective with a projection on latent states, and add image data augmentation for visual RL tasks.

AIS [71]: RP + ZP with detached ℓ_2 or forward KL in their approach, while RP + OP with 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 [71, 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 [63] 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 $\mathbb{P}_{\phi,\theta}(z' \mid h, a)$ and $\mathbb{Q}_\phi(z' \mid h, a)$ be the predicted and real next latent distributions. The MMD with k_d can be reduced to ℓ_2 distance between the expectations of two distributions:

$$\begin{aligned}
&\mathrm{MMD}^2_{k_d}(\mathbb{P}_{\phi,\theta}, \mathbb{Q}_\phi; h, a) && (97)\\
&= -\mathbb{E}_{z'_1, z'_2 \sim \mathbb{P}_{\phi,\theta}}[d(z'_1, z'_2)] + 2\mathbb{E}_{z'_1 \sim \mathbb{P}_{\phi,\theta}, z'_2 \sim \mathbb{Q}_\phi}[d(z'_1, z'_2)] - \mathbb{E}_{z'_1, z'_2 \sim \mathbb{Q}_\phi}[d(z'_1, z'_2)] && (98)\\
&= -\mathbb{E}_{z'_1, z'_2 \sim \mathbb{P}_{\phi,\theta}}\left[\|z'_1 - z'_2\|_2^2\right] + 2\mathbb{E}_{z'_1 \sim \mathbb{P}_{\phi,\theta}, z'_2 \sim \mathbb{Q}_\phi}\left[\|z'_1 - z'_2\|_2^2\right] - \mathbb{E}_{z'_1, z'_2 \sim \mathbb{Q}_\phi}\left[\|z'_1 - z'_2\|_2^2\right] && (99)\\
&= 2\|\mathbb{E}_{z' \sim \mathbb{P}_{\phi,\theta}}[z' \mid h, a] - \mathbb{E}_{z' \sim \mathbb{Q}_\phi}[z' \mid h, a]\|_2^2 && (100)
\end{aligned}$$

Therefore, the MMD objective can be viewed as the expected ZP with ℓ_2 distance. They also propose forward KL to instantiate ZP loss. Nevertheless, AIS [71] do not show experiment results on learning ZP. Instead, they and the follow-up works [57, 64] implement AIS by learning OP loss with MMD objectives, resulting in learning *observation-predictive* representations. Another follow-up work, Discrete AIS [84], learns ZP loss with ℓ_2 objective in a discrete latent space, so that they can apply value iteration.

TD-MPC [26]: 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 trains the latent policy with MPC algorithm. They find that learning ZP works better than learning OR or not learning ZP in the DM Control suite.

TCRL [89]: RP + multi-step ZP with EMA cos. Temporal consistency reinforcement learning (TCRL) simplifies TD-MPC [26] 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.

ALM [16]: 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 $\phi(s)$ and the latent policy $\pi(z)$ 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, the 1-step objective for their encoder is computed as

$$\min_\phi -R_z(z_\phi, a) + D_{\mathrm{KL}}(\mathbb{P}_\theta(z' \mid z_\phi, a) \,||\, \mathbb{P}_\phi(z' \mid s')) - \mathbb{E}_{z' \sim \mathbb{P}_\theta(|z_\phi, a)}\left[Q^\pi(z', \pi(\overline{z'}))\right] \tag{101}$$

where $z_\phi \sim \mathbb{P}_\phi(z \mid s)$, and $R_z(z, a)$ 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 [28] for policy optimization with a planning horizon of 3 steps.

Successor Representations and Features [2, 45]: ϕ_{Q^*} + RP + weak ZP. Here, we introduce successor features (SF) with our notation. Suppose the expected reward function can be computed as

$$\mathbb{E}[r \mid s, a] = g(\phi(s), a)^\top w, \quad \forall s, a \tag{102}$$

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[8]. In our notation, Eq. 102 is RP condition for ϕ.

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. 102. This special case is known as **successor representation** (SR) setting [10]. In deep SR [38, 45], they allow learning ϕ with assuming $g(\phi(s), a) = \phi(s)$.

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

$$Q^\pi(s, a) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid S_0 = s, A_0 = a\right] \tag{103}$$

[8] 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.

$$= \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t g(\phi(s_t), a_t)^\top w \mid S_0 = s, A_0 = a \right] \tag{104}$$

$$= \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t g(\phi(s_t), a_t) \mid S_0 = s, A_0 = a \right]^\top w \tag{105}$$

$$:= \psi^\pi(s,a)^\top w \tag{106}$$

where $\psi^\pi(s,a)$ is called successor features [2], a geometric sum of future $g(\phi(s),a)$. Although ψ^π can belong to any function class, following deep SR [38, 45], 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. 106 in Bellman equation $Q^\pi(s,a) = \mathbb{E}_{s',a' \sim \pi}[R(s,a) + \gamma Q^\pi(s',a')]$, we have

$$f^\pi(\phi(s),a) = g(\phi(s),a) + \gamma \mathbb{E}_{s',a' \sim \pi}[f^\pi(\phi(s'),a')] \tag{107}$$

Therefore, Eq. 107 can be viewed as a **weak** version of ZP, because given any current latent state and action pair $(\phi(s),a)$, Eq. 107 can predict the expectation of some function of next latent state $\phi(s')$. ZP can imply Eq. 107 because it can predict exactly the distribution of next latent state.

With a combination of RP (Eq. 102), ϕ_{Q^*} (implied by Eq. 107 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) [45, 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 the expected ZP (not necessarily ZP), thus, still a **weak** version of bisimulation.

Following Lehnert and Littman [45], assume the finite latent space is composed of one-hot vectors: $\mathcal{Z} = \{e_1, e_2, \dots, e_n\}$, we can construct a matrix $F^\pi \in \mathbb{R}^{d \times n}$ with each column $F^\pi(i) = \mathbb{E}_{a \sim \pi(|e_i)}[F_a e_i]$.

$$\frac{1}{\gamma}(f^\pi(\phi(s),a) - g(\phi(s),a)) = \mathbb{E}_{s',a' \sim \pi}[f^\pi(\phi(s'),a')] \tag{108}$$

$$= \mathbb{E}_{s' \sim P(|s,a), a' \sim \pi(|\phi(s'))}[F_{a'}\phi(s')] \tag{109}$$

$$= \mathbb{E}_{s' \sim P(|s,a)}[F^\pi \phi(s')] = F^\pi \mathbb{E}_{s' \sim P(|s,a)}[\phi(s')] \tag{110}$$

By [45, 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.*, expected ZP holds.

C.2 Observation-Predictive Representations

Belief-Based Methods [21, 23, 25, 44]: RP + OR + ZP with online forward KL. As a major approach to solving POMDPs, belief-based methods extends belief MDPs [34] 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:

$$p(z_{1:T+1}, o_{1:T+1}, r_{1:T} \mid a_{1:T}) = p(z_1)p(o_1 \mid z_1) \prod_{t=1}^{T} p(r_t \mid z_t, a_t) p(z_{t+1} \mid z_t, a_t) p(o_{t+1} \mid z_{t+1}) \tag{111}$$

$$q(z_{1:T+1} \mid h_{T+1}) = \prod_{t=0}^{T} q(z_{t+1} \mid h_{t+1}) = \prod_{t=0}^{T} q(z_{t+1} \mid z_t, a_t, o_{t+1}) \tag{112}$$

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

$$\mathbb{E}_{h_{T+1}, r_{1:T}}[\log p_\theta(o_{1:T+1}, r_{1:T} \mid a_{1:T})] \tag{113}$$

$$= \mathbb{E}_{h_{T+1}, r_{1:T}} \left[\log \mathbb{E}_{q(z_{1:T+1} | h_{T+1})} \left[\frac{p(z_{1:T+1}, o_{1:T+1}, r_{1:T} \mid a_{1:T})}{q(z_{1:T+1} \mid h_{T+1})} \right] \right] \tag{114}$$

$$\geq \mathbb{E}_{h_{T+1}, r_{1:T}, z_{1:T} \sim q(\cdot | h_{T+1})} \left[\log \frac{p(z_{1:T}, o_{1:T+1}, r_{1:T} \mid a_{1:T})}{q(z_{1:T+1} \mid h_{T+1})} \right] \tag{115}$$

$$= \mathbb{E}_{h_{T+1}, r_{1:T}, z_{1:T+1} \sim q(h_{T+1})} \left[\sum_{t=0}^{T} \underbrace{\log p(o_{t+1} \mid z_{t+1})}_{(1)} + \underbrace{\log p(r_t \mid z_t, a_t)}_{(2)} - \underbrace{\log \frac{q(z_{t+1} \mid h_{t+1})}{p(z_{t+1} \mid z_t, a_t)}}_{(3)} \right] \tag{116}$$

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 $\mathbb{E}_{h_{t+1}}[D_{\mathrm{KL}}(q(z_{t+1} \mid h_{t+1}) \,||\, p(z_{t+1} \mid z_t, a_t))]$, which is exactly the forward KL objective to learn ZP. From our relation graph (Fig. 1; Prop. 5), 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,

$$\mathbb{I}(o'; z') = \mathbb{E}_{o', z' \sim P(o', z')} \left[\log \frac{P(o', z')}{P(o') P(z')} \right] \tag{117}$$

$$= \mathbb{E}_{o', z' \sim P(o', z')} \left[\log \frac{P(o' \mid z')}{P(o')} \right] \tag{118}$$

$$= \mathbb{E}_{o', z' \sim P(o', z')} [\log P(o' \mid z')] + \mathbb{H}(P(o')) \tag{119}$$

$$= \mathbb{E}_{h, a, o' \sim P(|h, a), z' \sim P(|h')} [\log P(o' \mid z')] + \mathbb{H}(P(o')) \tag{120}$$

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 [56]: ϕ_{Q^*} + OP. Online Feature Extractor Network (OFENet) trains the state encoder using an auxiliary task of OP loss with ℓ_2 distance. They show strong performance of their approach over model-free baseline in standard MuJoCo benchmark. Follow-up work [41] empirically find that ϕ_{Q^*} + RP slightly improves up model-free RL, but much worse than ϕ_{Q^*} + OP in MuJoCo benchmark.

SAC-AE [85]: 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. 6), SAC-AE also approximates observation-predictive representation.

PSR [47] and belief trajectory equivalence [6]: Rec + multi-step OP and RP. Predictive State Representation (PSR) aims to learn a history encoder ϕ and transition model P_O such that

$$P(o_{t+1:t+k} \mid h_t, a_{t:t+k-1}) = P_O(o_{t+1:t+k} \mid \phi(h_t), a_{t:t+k-1}), \quad \forall h, a, o \tag{121}$$

which implies multi-step OP (defined in Prop. 9) in POMDPs. The original PSR uses linear transition models. Follow-up work on PSRs [31] and belief trajectory equivalence introduce multi-step RP to PSR. In Castro et al. [6], they show that single-step OP and RP do not necessarily imply multi-step OP and RP in POMDPs, summarized in Prop. 9. In this sense, PSR is a stronger notion of belief abstraction.

Causal state representations [87]: Rec + OP + RP. This work connects observation-predictive representations in POMDPs with causal state models in computational mechanics [65]. Specifically, they show that belief trajectory equivalence (Rec + multi-step OP and RP) [6] 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.

C.3 Other Related Representations

UNREAL [30], Loss is its own Reward [66]. 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

$$\exists P_{\text{inv}} : \mathcal{Z} \times \mathcal{Z} \to \Delta(\mathcal{A}), \quad s.t. \quad P_{\text{inv}}(a \mid \phi(s), \phi(s')) = P(a \mid s, s'), \quad \forall s, a, s' \tag{122}$$

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

VPN [54], MuZero [61]: 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 MCTS algorithm.

E2C [82] and World Model [20]: 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 [43], DRIML [48], ContraBAR [8]): ϕ_{Q^*} (RP) + weak OP (OR). CURL (ϕ_{Q^*} + weak OR) introduces contrastive learning using the infoNCE objective [55] 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 [59]. In MDPs, it is a lower bound of $\mathbb{I}(s; z)$, which correspond to OR objectives (Eq. 117). 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 [60], by data processing inequality:

$$\mathbb{I}(z'; z, a) \leq \mathbb{I}(z'; s, a) \leq \mathbb{I}(s'; s, a) \tag{123}$$

$$\mathbb{I}(z'; z, a) \leq \mathbb{I}(s'; z, a) \leq \mathbb{I}(s'; s, a) \tag{124}$$

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 [53]. 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 [8, Theorem 4.3], under certain optimality condition, the objective can lead to learning RP and OP conditions.

Learning Markov State Abstraction [1]: ϕ_{Q^*} + ZM. From Prop. 3, 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 [7]: 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

$$U_\phi(x, y) = \frac{1}{2}(\|\phi(x)\|_2^2 + \|\phi(y)\|_2^2) + \beta \arctan(\sqrt{1 - \cos(\phi(x), \phi(y))^2}, \cos(\phi(x), \phi(y))) \tag{125}$$

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

SALE [14]: ZP with detached ℓ_2. TD7 algorithm contains state-action learned embeddings (SALE) method that learns a state encoder to predict next latent states in MDPs. They conduct extensive ablation studies in the MuJoCo benchmark. They detach 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 train the encoder *solely* with ZP loss, because they find that this is slightly better than training with ZP + RP, and much better than end-to-end training (ZP + ϕ_Q^*). Lastly, both raw states s and (detached) latent states z are inputted into the actor-critic, maintaining original information for optimal decision-making.

D Experimental Details

D.1 Small Scale Experiments to Illustrate Theorem 3

In this section, we discuss the details of the experiment 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:

$$\left\| \begin{bmatrix}\phi^\top S & A\end{bmatrix} \begin{bmatrix}\theta_z \\ \theta_a\end{bmatrix} - \widetilde{\phi}^\top S' \right\|_2, \tag{126}$$

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 [5] `jax.numpy.linalg.lstsq` method when using `float32` encoding.

Mountain car [50]. We follow the dynamics and parameters used in [74, 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 [49]. 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.