Provable Representation with Efficient Planning for Partially Observable Reinforcement Learning

Hongming Zhang[*1] **Tongzheng Ren**[*2] **Chenjun Xiao**[3] **Dale Schuurmans**[4 1] **Bo Dai**[4 5]

Abstract

In most real-world reinforcement learning applications, state information is only partially observable, which breaks the Markov decision process assumption and leads to inferior performance for algorithms that conflate observations with state. Partially Observable Markov Decision Processes (POMDPs), on the other hand, provide a general framework that allows for partial observability to be accounted for in *learning, exploration and planning*, but presents significant computational and statistical challenges. To address these difficulties, we develop a representation-based perspective that leads to a coherent framework and tractable algorithmic approach for practical reinforcement learning from partial observations. We provide a theoretical analysis for justifying the statistical efficiency of the proposed algorithm, and also empirically demonstrate the proposed algorithm can surpass state-of-the-art performance with partial observations across various benchmarks, advancing reliable reinforcement learning towards more practical applications.

1. Introduction

Reinforcement learning (RL) addresses the problem of making sequential decisions that maximize a cumulative reward through interaction and observation in an environment (Mnih et al., 2013; Levine et al., 2016). The Markov decision process (MDP) has been the standard mathematical model used for most RL algorithm design. However, the success of MDP-based RL algorithms (Zhang et al., 2022; Ren et al., 2023c) relies on an assumption that state information is fully observable, which implies that the optimal policy is memoryless, *i.e.*, optimal actions can be selected based purely on the current state (Puterman, 2014). However, such an assumption typically does not hold in practice. For example, in chatbot learning (Jiang et al., 2021) or video game control (Mnih et al., 2013) only dialogue exchanges or images are observed, from which state information only can be partially inferred. The violation of full observability can lead to significant performance degeneration of MDP-based RL algorithms in such scenarios.

The Partially Observed Markov Decision Process (POMDP) (Åström, 1965) has been proposed to extend the classical MDP formulation by introducing observation variables that only give partial information about the underlying latent state (Hauskrecht & Fraser, 2000; Roy & Gordon, 2002; Chen et al., 2016). This extension greatly expands the practical applicability of POMDPs over MDPs, but the additional uncertainty over the underlying state given only observations creates a non-Markovian dependence between successive observations, even though Markovian dependence is preserved between latent states. Consequently, the optimal policy for a POMDP is no longer memoryless but *entire-history* dependent, expanding the state complexity exponentially w.r.t. horizon length. Such a non-Markovian dependence creates significant computational and statistical challenges in *planning*, and in *learning* and *exploration*. In fact, without additional assumptions, computing an optimal policy for a POMDP with known dynamics (*i.e.*, planning) is PSPACE-complete (Papadimitriou & Tsitsiklis, 1987), while the sample complexity of learning for POMDPs grows exponentially w.r.t. the horizon (Jin et al., 2020a).

Despite the worst case hardness of POMDPs, given their importance in practice, there has been extensive work on developing practical RL algorithms that can cope with partial observations. One common heuristic is to extend MDP-based RL algorithms by maintaining a history window over observations to encode a policy or value function, *e.g.*, recurrent neural networks (Wierstra et al., 2007; Hausknecht & Stone, 2015; Zhu et al., 2017). Such algorithms have been applied to many real-world applications with image- or text-based observations (Berner et al., 2019; Jiang et al., 2021), sometimes even surpassing human-level performance (Mnih et al., 2013; Kaufmann et al., 2023).

These empirical successes have motivated investigation into *structured* POMDPs that allow some of the core compu-

[*]Equal contribution [1]University of Alberta [2]UT Austin [3]The Chinese University of Hong Kong, Shenzhen [4]Google DeepMind [5]Georgia Tech. Correspondence to: Bo Dai <bodai@google.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

tational and statistical complexities to be overcome, and provides an improved understanding of exploitable structure and practical new algorithms with rigorous justification. For example, the concept of *decodability* has been used to express POMDPs where the latent state can be exactly recovered from a window of *past* observations (Efroni et al., 2022; Guo et al., 2023). *Observability* is another special structure, where the m-step emission model is assumed to be full-rank, allowing the latent state to be identified from m *future* observation sequences (Jin et al., 2020a; Golowich et al., 2022; Liu et al., 2022; 2023). Such structures eliminate unbounded history dependence, and thus, reduce the computational and statistical complexity. However, most works rely on the existence of an ideal computational oracle for planning, which, unsurprisingly, is infeasible in most cases, hence difficult to apply in practice. Although there have been a few attempts to overcome the computational complexity of POMDPS, these algorithms are either only applicable to the tabular setting (Golowich et al., 2022) or rely on an integration oracle that quickly become intractable for large observation spaces (Guo et al., 2018). This gap immediately motivates the question:

Can **efficient** *and* **practical** *RL algorithms be designed for partial observations by exploiting natural structures?*

By "efficient" we mean the statistical complexity avoids an exponential dependence on history length, while by "practical" we mean that every component of *learning, planning* and *exploration* can be easily implemented and applied in practical settings. In this paper, we provide an **affirmative** answer to this question. More specifically,

- We reveal for the first time that an L-decodable POMDP admits a sufficient representation, the *Multi-step Latent Variable Representation (μLV-Rep)*, that supports exact and tractable linear representation of the value functions (Section 4.1), breaking the fundamental computational barriers explained in more detail in Section 3.
- We design a computationally efficient planning algorithm that can implement both the principles of optimism and pessimism in the face of uncertainty for online and offline POMDPs respectively, by leveraging the learned sufficient representation μLV-Rep (Section 4.2).
- We provide a theoretical analysis of the sample complexity of the proposed algorithm, justifying its efficiency in balancing exploitation versus exploration in Section 5.
- We conduct a comprehensive empirical comparison to current existing RL algorithms for POMDPs on several benchmarks, demonstrating the superior empirical performance of μLV-Rep (Section 7).

2. Preliminaries

We follow the definition of a POMDP given in (Efroni et al., 2022; Liu et al., 2022; 2023), which is formally denoted as a tuple $\mathcal{P} = (\mathcal{S}, \mathcal{A}, \mathcal{O}, r, H, \rho_0, \mathbb{P}, \mathbb{O})$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, and $\mathcal{O}$ is the observation space. The positive integer H denotes the horizon length, ρ_0 is the initial state distribution, $r : \mathcal{O} \times \mathcal{A} \to [0, 1]$ is the reward function, $\mathbb{P}(\cdot|s, a) : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition kernel capturing dynamics over latent states, and $\mathbb{O}(\cdot|s) : \mathcal{S} \to \Delta(\mathcal{O})$ is the emission kernel, which induces an observation from a given state.

Initially, the agent starts at a state s_0 drawn from $\rho_0(s)$. At each step h, the agent selects an action a from $\mathcal{A}$. This leads to the generation of a new state s_{h+1} following the distribution $\mathbb{P}(\cdot|s_h, a_h)$, from which the agent observes o_{h+1} according to $\mathbb{O}(\cdot|s_{h+1})$. The agent also receives a reward $r(o_{h+1}, a_{h+1})$. Observing o instead of the true state s leads to a non-Markovian transition between observations, which means we need to consider policies $\pi_h : \mathcal{O} \times (\mathcal{A} \times \mathcal{O})^h \to \Delta(\mathcal{A})$ that depend on the entire history, denoted by $\tau_h = \{o_0, a_0, \cdots, o_h\}$. Let $[H] := \{0, \ldots, H\}$. Then the value associated with policy $\pi = \{\pi_h\}_{h\in[H]}$ is defined as $v^\pi = \mathbb{E}_\pi \left[\sum_{h\in[H]} r(o_h, a_h)\right]$. The goal is to find the optimal policy $\pi^* = \arg\max_\pi v^\pi$. Note that the MDP given by $\mathcal{M} = (\mathcal{S}, \mathcal{A}, r, H, \rho_0, \mathbb{P})$ is a special case of a POMDP, where the state space $\mathcal{S}$ is equivalent to the observation space $\mathcal{O}$, and the emission kernel $\mathbb{O}(o|s)$ is defined as $\delta(o = s)$.

Define the belief function $b(\cdot) : \mathcal{O}\times(\mathcal{A}\times\mathcal{O})^h \to \Delta(\mathcal{S})$. Let $b(s_1|o_1) = \mathbb{P}(s_1|o_1)$. Then we can recursively compute:

$$b(s_{h+1}|\tau_{h+1}) \propto \int_{\mathcal{S}} b(s_h|\tau_h)\mathbb{P}(s_{h+1}|s_h, a_h)\mathbb{O}(o_{h+1}|s_{h+1})ds_h. \tag{1}$$

With such a definition, one can convert a POMDP to an equivalent MDP over beliefs, denoted as $\mathcal{M}_b = (\mathcal{B}, \mathcal{A}, R_h, H, \mu_b, T_b)$, where $\mathcal{B} \subseteq \Delta(\mathcal{S})$ represents the set of possible beliefs, $\mu_b(\cdot) = \int b(\cdot|o_1)\mu(o_1)do_1$, and

$$\mathbb{P}_b\left(b_{h+1}|b_h, a_h\right) = \int \mathbf{1}_{b_{h+1}=b(\tau_h,a_h,o_{h+1})}\mathbb{P}(o_{h+1}|b_h, a_h)do_{h+1}. \tag{2}$$

Notice that $b(\cdot) \in \mathcal{B}$ is a mapping $b : \mathcal{O} \times (\mathcal{A} \times \mathcal{O})^h \to \Delta(\mathcal{S})$, *i.e.*, each belief corresponds to a density measure over the state space, thus $\mathbb{P}_b\left(\cdot|b_h, a_h\right)$ represents a conditional operator that characterizes the transition between belief distributions. For a given policy $\pi : \mathcal{B} \to \Delta(\mathcal{A})$, we can define the state value function $V_h^\pi(b_h)$ and state-action value function $Q_h^\pi(b_h, a_h)$ respectively for the belief MDP (*i.e.*, equivalently for the original POMDP) as:

$$V_h^\pi(b_h) = \mathbb{E}\left[\sum_{t=h}^H r(o_t, a_t)|b_h\right],$$
$$Q_h^\pi(b_h, a_h) = \mathbb{E}\left[\sum_{t=h}^H r(o_t, a_t)|b_h, a_h\right]. \tag{3}$$

Therefore, the Bellman equation can be expressed as

$$V_h^\pi(b_h) = \mathbb{E}_\pi\left[Q_h^\pi(b_h, a_h)\right],$$
$$Q_h^\pi(b_h, a_h) = r(o_h, a_h) + \mathbb{E}_{\mathbb{P}_b}\left[V_{h+1}^\pi(b_{h+1})\right]. \tag{4}$$

Although this reformulation of a POMDP provides a re-

duction to an equivalent MDP, it is crucial to recognize that the equivalent MDP is based on beliefs, which are also not directly observed. More importantly, these beliefs are densities that depend on the *entire history*, hence the joint distribution is supported on a space of growing dimension, leading to exponential representation complexity even when the number of states is finite. These difficulties result in infeasible computational and statistical complexity (Papadimitriou & Tsitsiklis, 1987; Jin et al., 2020a), and reveal the inherent suboptimality of directly applying MDP-based RL algorithms in a POMDP. Moreover, incorporating function approximation in learning and planning is significantly more involved in a POMDP than an MDP.

Consequently, several special structures have been investigated in the theoretical literature to reduce the statistical complexity of learning in a POMDP. Specifically, *L-decodability* and *γ-observability* have been introduced in (Du et al., 2019; Efroni et al., 2022) and (Golowich et al., 2022; Even-Dar et al., 2007) respectively.

Definition 1 (L-decodability (Efroni et al., 2022))**.** *$\forall h \in [H]$, define*

$$x_h \in \mathcal{X} := (\mathcal{O} \times \mathcal{A})^{L-1} \times \mathcal{O},$$
$$x_h = (o_{h-L+1}, a_{h-L+1}, \cdots, o_h). \tag{5}$$

A POMDP is L-decodable if there exists a decoder $p^* : \mathcal{X} \to \Delta(\mathcal{S})$ such that $p^*(x_h) = b(\tau_h)$.

Definition 2 (γ-observability (Golowich et al., 2022; Even–Dar et al., 2007))**.** *Denote $\langle \mathbb{O}, b \rangle := \int \mathbb{O}_h(\cdot|s)\, b(s)\, ds$, for arbitrary beliefs b and b' over states. A POMDP is γ-observable if $\|\langle \mathbb{O}, b\rangle - \langle \mathbb{O}, b'\rangle\|_1 \geqslant \gamma \|b - b'\|_1$.*

Note that we slightly generalize the decodability definition of Efroni et al. (2022), which assumes $b(\tau_h)$ is a Dirac measure on $\mathcal{S}$. It is worth noticing that γ-observability and L-decodability are highly related. Existing works have shown that a γ-observable POMDP can be well approximated by a decodable POMDP with a history of suitable length L (Golowich et al., 2022; Uehara et al., 2022; Guo et al., 2023) (see Appendix B for a detailed discussion). Hence, in the main text, we focus on L-decodable POMDPs, which can be directly extended to γ-observable POMDPs.

Although these structural assumptions have been introduced to ensure the sample complexity of learning in a structured POMDP is reduced, the *computational* tractability of planning and exploring given such structures remains open.

3. Difficulties in Learning with POMDPs

Before attempting to design practical RL algorithms for structured POMDPs, we first demonstrate the generic difficulties of learning in POMDPs, including the difficulty of planning even in a *known* POMDP. These challenges only become more difficult when combined with the need to *explore* in an unknown POMDP during learning.

The major difficulty of **planning in a known POMDP** lies in evaluating the value function (Ross et al., 2008), which is necessary for both temporal-difference and policy gradient based planning and learning methods. For clarity in explaining the difficulties, consider estimating Q^π for a given policy π. (The same difficulties arise in estimating Q^*.) Based on the Bellman equation (4), we have: $Q_h^\pi(b_h, a_h) = r(o_h, a_h) + \mathbb{E}_{\mathbb{P}(o_{h+1}|b_h,a_h)}\left[V_{h+1}^\pi(b_{h+1})\right]$. To execute dynamic programming in the hth time step, *i.e.*, to determine Q_h^π, we require several components:

i) The belief $b_{h+1}(\cdot)$ must be calculated, as defined in (1), which requires marginalization and renormalization, with a computational complexity proportional to the number of states. For continuous control, the calculation involves integration, which is intractable in general.

ii) Representing the value function $V_{h+1}^\pi(\cdot)$ quickly becomes intractable, even for discrete state and action spaces. In general, expressing $V_{h+1}^\pi(\cdot)$ requires space proportional to size of the support of beliefs, which is exponential w.r.t. the horizon. (The optimal value function $V_{h+1}^*(\cdot)$ can be represented by convex piece-wise linear function (Smallwood & Sondik, 1973), but the number of components can grow exponentially in the horizon.) For continuous state POMDPS, no compact characterization is yet known for the space of value functions.

iii) The expectation $\mathbb{E}_{\mathbb{P}(o_{h+1}|b_h,a_h)}\left[V^\pi(b_{h+1})\right]$ requires integration w.r.t. the distribution $\mathbb{P}(o_{h+1}|b_h, a_h) = \int \mathbb{O}(o_{h+1}|s_{h+1})\mathbb{P}(s_{h+1}|s_h, a_h)\, b_h(s_h)\, ds_h ds_{h+1}$ with computational cost proportional to the size of the support of beliefs, which is also exponential in the horizon.

These calculations are generally intractable in a POMDP, particularly one with a continuous state space. Due to this intractability, the exact calculation of the value function via the Bellman equation is generally infeasible. Consequently, many approximation methods have been proposed for each component of planning (Ross et al., 2008). However, the approximation error in each step can be amplified when these steps are composed through the Bellman recursion, leading to significantly suboptimal performance.

Due to promising theoretical results that exploit γ-observability or L-decodability to reduce the exponential dependence on horizon in terms of statistical complexity, there have been many attempts to exploit these properties to also achieve computationally tractable planning for RL in a POMDP. Golowich et al. (2022) construct an approximate MDP for an L-observable POMDP, whose approximation error can be characterized for planning. However, the proposed algorithm only works for POMDPs with discrete observations, states and actions. Alternatively, Guo et al. (2023) exploits low-rank latent dynamics in an L-decodable POMDP, which induces a *nonlinear* representation for value functions. Unfortunately, this nonlinear structure has not been shown to support any practical algorithm. To imple-

ment the backup step in dynamic programming for Q^π, the algorithm in (Guo et al., 2023) requires an integration-oracle to handle the nonlinearity, which is generally infeasible.

Another barrier to practical learning in a POMDP is **exploration**, which is usually implemented in planning by appealing to the principle of optimism in the face of uncertainty (OFU). Unfortunately, OFU usually requires not only the estimation of the model or value function, but also their uncertainty, *e.g.*, a confidence interval in frequentist approaches (Liu et al., 2022; 2023) or the posterior in Bayesian approaches (Simchowitz et al., 2021). Practical and general implementations are not yet known in either case.

These major issues, which have not been carefully considered in empirical RL algorithms that leverage history-based neural networks (Wierstra et al., 2007; Hausknecht & Stone, 2015; Zhu et al., 2017), interact in ways that exaggerate the difficulty of developing efficient algorithms for POMDPs.

4. Multi-step Latent Variable Representation

In this section, we develop the first **computationally efficient** planning and exploration algorithm by further leveraging the structure of an L-decodable POMDP. The main contribution is to identify a latent variable representation that can *linearly represent* the value function of an arbitrary policy in an L-step decodable POMDP. From this representational perspective, we show that explicit belief calculation can be bypassed and the backup step can be directly calculated via optimization. We then show how a latent variable representation learning algorithm can be designed, based upon (Ren et al., 2023b), that leads to the proposed *Multi-step Latent Variable Representation (μLV-Rep)* method.

4.1. Efficient Policy Evaluation from Key Observations

Although the equivalent belief MDP provides a Markovian Bellman recursion (4), as discussed in Section 3, this belief viewpoint hampers tractability in planning. In the following, we introduce the key observations that allow us to resolve the aforementioned difficulties in planning step-by-step.

Belief Elimination. Our first key observation is that L-decodability eliminates an aspect of exponential complexity:

- *In an L-decodable POMDP, it is sufficient to recover the belief state by an L-step memory x_h rather than the entire history τ_h.*

This is exactly the definition of L-decodability given in Definition 1. From this observation that beliefs $b_h(s)$ can be represented with a decoder from L-step windows x_h (5), we obtain the simplification $Q_h^\pi(b_h(\tau_h), a_h) = Q_h^\pi(b_h(x_h), a_h)$. In addition, L-decodability also helps reduce dependence in the dynamics, leading to $\mathbb{P}(o_{h+1}|b_h(\tau_h), a_h) = \mathbb{P}(o_{h+1}|b_h(x_h), a_h)$. This outcome reduces the statistical complexity, as previously exploited in (Efroni et al., 2022; Guo et al., 2023). However, it is not sufficient to ensure computationally friendly planning.

Next we make a second key observation that allows us to eliminate the explicit use of beliefs in the Bellman equation:

- *An observation-based value function and transition model bypasses the necessity for belief computation.*

That is, since $b_h(\cdot)$ is a mapping from the history x_h to the space of probability densities over state, we can reparametrize the composition $Q_h^\pi(b_h(x_h), a_h)$ as $Q_h^\pi(x_h, a_h)$, which is a function directly over (x_h, a_h). Similarly, we can also consider the transition dynamics directly defined over the observation history, $\mathbb{P}(o_{h+1}|x_h, a_h)$, rather than work with the far more complex dynamics between successive beliefs. With such a reformulation, we avoid explicit dependence on beliefs in the Q^π-function and dynamics in the Bellman recursion (4). This resolves the first difficulty that arises from explicit belief calculation, leading to the simplified Bellman equation:

$$Q_h^\pi(x_h, a_h) = r(o_h, a_h) + \mathbb{E}_{\mathbb{P}^\pi(o_{h+1}|x_h,a_h)}\left[V_{h+1}^\pi(x_{h+1})\right]. \quad (6)$$

Linear Representation for Q^π. Next, we address a second key difficulty that arises in representing $V_{h+1}^\pi(x_{h+1}) = \mathbb{E}_{a\sim\pi}\left[Q_{h+1}^\pi(x_{h+1}, a_{h+1})\right]$. We seek an exact representation that can be recursively maintained. Inspired by the success of representing value functions exactly in linear MDPs (Jin et al., 2020b; Yang & Wang, 2020), it is natural to consider x_h as a mega-state (Efroni et al., 2022) and factorize the observation distribution in (6) as $\mathbb{P}(o_{h+1}|b_h, a_h) = \langle \phi(x_h, a_h), \mu(o_{h+1})\rangle$ over basis functions ϕ and μ. However, it is important to realize there is an additional dependence of $V_{h+1}^\pi(x_{h+1})$ on (x_h, a_h), since

$$x_{h+1} = (o_{h-L+2}, a_{h-L+2}, \cdots, o_h, a_h, o_{h+1})$$

has an overlap with

$$(x_h, a_h) = (o_{h-L+1}, a_{h-L+1}, o_{h-L+2}, a_{h-L+2}, \cdots, o_h, a_h).$$

This overlap breaks the maintained linear structure, since

$$Q_h^\pi(b_h, a_h) = r_h(o_h, a_h) + \left\langle \phi(x_h, a_h), \underbrace{\int \mu(o_{h+1}) V_{h+1}(x_h, a_h, o_{h+1})\, do_{h+1}}_{w(x_h,a_h)} \right\rangle.$$

That is, the nonlinearity in $w(x_h, a_h)$ prevents the direct extension of linear MDPs with mega-states (Efroni et al., 2022) to the POMDP case (Guo et al., 2023).

Nevertheless, a linear representation for the value function can still be recovered if the dependence of the $h+1$-step value function on (x_h, a_h) can be eliminated. The next key observation shows how this overlap might be avoided:

- *By L-step decodability, $V_{h+L}^\pi(x_{h+L})$ is independent of (x_h, a_h).*

This observation inspires us to consider the L-*step* Bellman equation for $Q_h^\pi(\tau_h, a_h)$, which can be easily derived by expanding (4) forward in time to:

$$Q_h^\pi(x_h, a_h) = \mathbb{E}^\pi_{x_{h+1:h+L}|x_h,a_h}\left[\sum_{i=h}^{h+L-1} r(o_i, a_i) + V_{h+L}^\pi(x_{h+L})\right]. \quad (7)$$

At first glance, the L-step forward expansion only involves $V^{\pi}_{h+L}(x_{h+L})$, which would seem to eliminate the overlapping dependence of x_{h+L} on (x_h, a_h) due to L-step decodability. However, the forward steps are conducted according to the policy $\pi_{h+1:h+L-1}$, which still depends on parts of (x_h, a_h), hence the distribution $\mathbb{P}^{\pi}(x_{h+L}|x_h, a_h)$ in the expectation in (7) still retains a dependence on (x_h, a_h).

We introduce the final key observation that allows us to fully eliminate the dependence:

- *For any policy π, there exists a corresponding policy ν_{π} that conditions on a sufficient latent variable to generate the same expected observation dynamics while being independent of history older than L steps.*

The ν_{π} is known as the *"moment matching policy"* (Efroni et al., 2022). The existence of such an equivalent ν^{π} is guaranteed by L-decodability: we defer the detailed construction of ν_{π} to Appendix C to avoid distraction, and focus on algorithm design in the main text.

With such a policy ν_{π}, the dependence of $\pi_{h+1:h+L-1}$ on (x_h, a_h) at the Lth-step in $\mathbb{P}^{\pi}(x_{h+L}|x_h, a_h)$ can be eliminated. Specifically, consider

$$\begin{aligned}\mathbb{P}^{\pi}(x_{h+L}|x_h, a_h) &= \int p(z_{h+1}|x_h, a_h)\mathbb{P}^{\nu_{\pi}}(x_{h+L}|z_{h+1})\, dz_{h+1} \\ &= \langle p(\cdot|x_h, a_h), \mathbb{P}^{\nu_{\pi}}(x_{h+L}|\cdot)\rangle_{L_2(\mu)}, \end{aligned} \tag{8}$$

where z denotes the latent variable and the first equality follows from the construction of ν_{π}. We emphasize that the idea of the moment matching policy was only considered as a proof technique in (Efroni et al., 2022), and not previously exploited to uncover linear structure for algorithm design.

Remark (Identifiability): It should be noted that we deliberately use a latent variable z rather than s in (8), to emphasize the learned latent variable structure can be different from the ground truth state, thus avoiding an *identifiability* assumption. Nevertheless, the learned structure has the same effect in representing Q^{π} linearly.

Based on the above observations, we have addressed the necessary components for a linear representation of the value function, achieved by introducing (8) into (7). For the first term in (7), for $\forall k \in \{1, \ldots, L-1\}$, we have

$$\begin{aligned}&\mathbb{E}^{\pi}_{o_{h+k}|x_h, a_h}\left[r(o_{h+k}, a_{h+k})\right] \\ &= \left\langle p(\cdot|x_h, a_h), \underbrace{\int \mathbb{P}^{\nu_{\pi}}(o_{h+k}, a_{h+k}|\cdot)\, r(o_{h+k}, a_{h+k})\, do_{h+k} da_{h+k}}_{w^{\pi}_k(\cdot)} \right\rangle.\end{aligned} \tag{9}$$

That is, using the "moment matching policy" trick, the policy ν_{π} is independent of (x_h, a_h), while $\mathbb{P}^{\nu_{\pi}}(o_{h+k}, a_{h+k}|\cdot)$ is independent of history, leading to the linear representation in (9). Similarly, for the second term in (7), we have

$$\begin{aligned}&\mathbb{E}_{\pi}\left[V^{\pi}_{h+L}(x_{h+L})\right] = \int \mathbb{P}^{\pi}(x_{h+L}|x_h, a_h)\, V^{\pi}(x_{h+L})\, dx_{h+L} \\ &= \left\langle p(\cdot|x_h, a_h), \underbrace{\int \mathbb{P}^{\nu_{\pi}}(x_{h+L}|\cdot)\, V^{\pi}(x_{h+L})\, dx_{h+L}}_{w^{\pi}_{h+L}(\cdot)} \right\rangle.\end{aligned} \tag{10}$$

Recall that x_{h+L} does not overlap with x_h, so under ν^{π}, $w^{\pi}_{h+L}(\cdot)$ is independent of (x_h, a_h).

By combining (9) and (10) and defining $w^{\pi} = \sum_{k=h}^{h+L} w^{\pi}_{h+k}$, we conclude that, in an L-step decodable POMDP, the value function Q^{π} can be represented linearly in $p(\cdot|x_h, a_h)$ as

$$Q^{\pi}_h(x_h, a_h) = \langle p(\cdot|x_h, a_h), w^{\pi}(\cdot)\rangle_{L_2(\mu)}, \tag{11}$$

under the assumption that $r(o_h, a_h) = \langle p(\cdot|x_h, a_h), \omega^r(\cdot)\rangle$, which can be easily achieved by feature augmentation (Ren et al., 2023a).

Least Square Policy Evaluation. The final difficulty in a practical algorithm design is addressing the expectation calculation in the Bellman equation (7). With the linear representation established for the value function Q^{π}_h, the entire backup step for dynamic programming with the Bellman recursion can be replaced by a least squares regression in the space spanned by $\phi(x_h, a_h)$. Specifically, at step h, we seek an estimate of $Q^{\pi}_{h+L}(x_{h+L}, a_{x+L})$ that is expressed as $\langle \tilde{w}^{\pi}_{h+L}, p(\cdot|x_{h+L}, a_{a+L})\rangle$, which can be obtained by the optimization:

$$\begin{aligned}\min_{w^{\pi}_h} \mathbb{E}^{\pi}_{x_{h+1:h+L}, x_h, a_h}\Bigg[\Bigg(&\langle w^{\pi}_h, p(\cdot|x_h, a_h)\rangle - \\ &\Bigg(\sum_{i=h}^{h+L-1} r(o_i, a_i) + \langle \tilde{w}^{\pi}_{h+L}, p(\cdot|x_{h+L}, a_{h+L})\rangle\Bigg)\Bigg)^2\Bigg],\end{aligned} \tag{12}$$

This problem can be easily solved by stochastic gradient descent. Setting the gradient to zero, one obtains the optimal solution to (12) as:

$$\begin{aligned}&\langle w^{\pi}_h, p(\cdot|x_h, a_h)\rangle = \\ &\mathbb{E}_{x_{h+1:h+L}}\left[\sum_{i=h}^{h+L-1} r(o_i, a_i) + \langle \tilde{w}^{\pi}_{h+L}, p(\cdot|x_{h+L}, a_{h+L})\rangle\right],\end{aligned}$$

which completes the Bellman backup.

In summary, we obtain an efficient policy evaluation algorithm for L-decodable POMDPs that only requires **least squares optimization** upon a **linear representation** of an **observation-based** value function.

Remark (Connection to Linear MDPs (Jin et al., 2020b; Yang & Wang, 2020)): The linear structure we have revealed for POMDP value functions has some similarity to value function representations in linear MDPs. One key difference between the factorization in (8) and that in a linear MDP, *i.e.*, $\mathbb{P}(s'|s, a) = \langle \phi(s, a), \mu(s')\rangle$, is that the latter factorizes for different transitions of the representation. Therefore, in a linear MDP, one obtains a *policy-independent* decomposition, where both the components $\phi(s, a)$ and $\mu(s')$ from the transition dynamics are invariant w.r.t. the policy. By contrast, in (8) for a POMDP, one of the factors, $\mathbb{P}^{\nu_{\pi}}(x_{h+L}|\cdot)$, depends on the policy. Nevertheless, we have demonstrated that this does not affect the linear representation ability of $p(\cdot|x_h, a_h)$ for Q^{π}.

Remark (Connection to PSR (Littman & Sutton, 2001): Both the linear representation revealed in this paper and

Algorithm 1 Online Exploration for L-step decodable POMDPs with Latent Variable Representation

Input: Model Class $\mathcal{M} = \{\{(p_h(z|x_h, a_h), p_h(o_{h+1}|z)\}_{h\in[H]}\}$, Variational Distribution Class $\mathcal{Q} = \{\{q_h(z|x_h, a_h, o_{h+1})\}_{h\in[H]}\}$, Episode Number K.
Initialize $\pi_0^h(s) = \mathcal{U}(\mathcal{A}), \forall h \in [H]$ where $\mathcal{U}(\mathcal{A})$ denotes the uniform distribution on $\mathcal{A}$; $\mathcal{D}_{0,h} = \emptyset, \mathcal{D}'_{0,h} = \emptyset, \forall h \in [H]$.
for episode $k = 1, \cdots, K$ **do**
  Initialize $\mathcal{D}_{k,h} = \mathcal{D}_{k-1,h}, \mathcal{D}'_{k,h} = \mathcal{D}'_{k-1,h}$
  for Step $h = 1, \cdots, H$ **do**
    Collect the transition $(x_h, a_h, o_{h+1}, a_{h+1}, \cdots, o_{h+L-1}, a_{h+L-1}, o_{h+L})$ where $x_h \sim d_{\mathcal{P}}^{\pi_k,h}$, $a_{h:h+L-1} \sim \mathcal{U}(\mathcal{A})$, $o_{h+i} \sim \mathbb{P}^{\mathcal{P}}(\cdot|x_{h+i-1}, a_{h+i-1}), \forall i \in [L]$.
    $\mathcal{D}_{k,h} = \mathcal{D}_{k,h} \cup \{x_h, a_h, o_{h+1}\}$, $\mathcal{D}'_{k,h+i} = \mathcal{D}'_{k,h+i} \cup \{x_{h+i}, a_{h+i}, o_{h+i+1}\}, \forall i \in [L]$.
  end for
  Learn the latent variable model $\hat{p}_k(z|x_h, a_h)$ with $\mathcal{D}_{k,h} \cup \mathcal{D}'_{k,h}$ via maximizing the ELBO, and obtain the learned model $\widehat{\mathcal{P}}_k = \{(\hat{p}_{h,k}(z|x_h, a_h), \hat{p}_{h,k}(o_{h+1}|z))\}_{h\in[H]}$.
  (Optional) Set the exploration bonus $\hat{b}_{k,h}(s, a)$.
  Update policy $\pi_k = \arg\max_\pi V^\pi_{\widehat{\mathcal{P}}_k, r+\hat{b}_k}$.
end for
Return $\pi^1, \cdots, \pi^K$.

the predictive state representation (PSR) (Littman & Sutton, 2001) bypass explicit belief calculation by factorizing the observation transition system. However, there are significant differences between the two structures that affect planning and exploration. Specifically, the PSR is based on the assumption that, for any finite sequence of events $y_{h+1:k} = (o_{h+1:h+k}, a_{h:h+k-1})$, $k \in \mathbb{N}_+$, following a history x_h, the probability can be linearly factorized as $\mathbb{P}(o_{h+1:h+k}|x_h, a_{h:h+k-1}) = \langle \omega_{y_{h+1:k}}, \mathbb{P}(U|x_h)\rangle$, where $\omega_{y_{h+1:k}} \in \mathbb{R}^d$, $U := [u_i]_{i=1}^d$ is a set of core test events, and $\mathbb{P}(U|x_h)$ is the predictive state representation at step h. Then, the forward observation dynamics can be represented in a PSR via Bayes' rule, $\mathbb{P}(o_{h+2:k}|x_h, a_{h:h+k-1}, o_{h+1}) = \frac{\langle \omega_{y_{h+2:k}}, \mathbb{P}(U|x_h)\rangle}{\langle \omega_{y_{h+1}}, \mathbb{P}(U|x_h)\rangle}$, which introduces a *nonlinear* operation, making planning and exploration both extremely difficult.

4.2. Learning with Exploration

We have developed a linear representation for Q_h^π that enables efficient planning. This section discusses how to learn and explore on top of this representation. The full algorithm is presented in Algorithm 1.

Variational Learning of μLV-Rep. As we generally do not have the latent variable representation $p(\cdot|x_h, a_h)$ beforehand, it is essential to perform representation learning with online collected data. One straightforward idea is to apply maximum likelihood estimation on $\mathbb{P}^\pi(x_{h+k}|x_h, a_h)$. Although this is theoretically correct, due to the overlap between x_{h+k} and x_h, a naive parametrization unnecessarily wastes memory and computational costs. Recall that we only need $p(z_h|x_h)$ to represent Q_h^π, and from

$$p(o_{h+1:h+l}|x_h, a_h) = \int_{\mathcal{Z}} p(z_h|x_h, a_h) \cdot \underbrace{\prod_{i=1}^{l} \left[\int_{\mathcal{Z}} \mathbb{P}^\pi(z_{h+i}|z_{h+i-1}, a_i) p(o_{h+i}|z_{h+i}) dz_{h+i}\right]}_{\mathbb{P}^\pi(o_{h+1:h+l}|z_h)} dz_h, \tag{13}$$

we can obtain $p(\cdot|x_h, a_h)$ by performing maximum likelihood estimation (MLE) on $p(o_{h+1:h+l}|x_h, a_h)$ for arbitrary $l \in \mathbb{N}_+$. To obtain a tractable surrogate for the MLE of the latent variable model (13), we exploit the evidence lower bound (ELBO) (Ren et al., 2023b):

$$\begin{aligned}
&\log p(o_{h+1:h+l}|x_h, a_h) \\
&= \log \int_{\mathcal{Z}} p(z_h|x_h, a_h) \mathbb{P}^\pi(o_{h+1:h+l}|z_h) \\
&= \log \int_{\mathcal{Z}} \frac{p(z_h|x_h, a_h)\mathbb{P}^\pi(o_{h+1:h+l}|z_h)}{q(z|x_h, a_h, o_{h+1:h+l})} q(z|x_h, a_h, o_{h+1:h+l}) \\
&= \max_{q\in\Delta(\mathcal{Z})} \mathbb{E}_{q(\cdot|x_h, a_h, o_{h+1:h+l})} \left[\log \mathbb{P}^\pi(o_{h+1:h+l}|z_h)\right] \\
&\quad - D_{KL}\left(q(z|x_h, a_h, o_{h+1:h+l})||p(z_h|x_h)\right),
\end{aligned} \tag{14}$$

where the last equation comes from Jensen's inequality, with equality holding when $q(z|x_h, a_h, o_{h+1:h+l}) \propto p(z_h|x_h, a_h)\mathbb{P}^\pi(o_{h+1:h+l}|z_h)$. One can use (14) with data to fit the μLV-Rep. For the ease of the presentation, we choose $l = 1$ in Algorithm 1.

Practical Parametrization of Q^π with μLV-Rep. With μLV-Rep, we can represent $Q_h^\pi(x_h, a_h) = \langle p(z|x_h), w_h^\pi(z)\rangle_{L_2(\mu)}$. If the latent variable z in $p(z|x_h)$ is an enumeratable discrete variable, $Q^\pi(x_h, a_h) = \sum_{i=m} w^\pi(z_i) p(z_i|x_h)$, can be simply represented.

However, a discrete latent variable is not differentiable, which causes difficulty in learning. Therefore, we use a continuous latent variable z, which induces an infinite-dimensional $w(z)$. To address this challenge, we follow the trick in LV-Rep (Ren et al., 2023b) that forms $Q^\pi(x_h, z_h)$ as an expectation:

$$Q^\pi(x_h, a_h) = \langle p(z|x_h), w^\pi(z)\rangle = \mathbb{E}_{p(z|x_h)}\left[w^\pi(z)\right]$$

which can then be approximated by a Monte-Carlo method or random feature quadrature (Ren et al., 2023b):

$$Q^\pi(x_h, a_h) \approx \frac{1}{m}\sum_{i=1}^m w^\pi(z_i) \text{ or } \frac{1}{m}\sum_{i=1}^m \tilde{w}^\pi(\xi_i)\varphi(z_i, \xi_i)$$

with samples $z_i \sim p(z|x_h)$ and $\xi_i \sim P(\xi)$ as the random feature measure for the RKHS containing $w(z)$. Both approximations can be implemented by a neural network. Due to space limitations, we defer the derivation of the random feature quadrature to Appendix E.

Planning and Exploration with μLV-Rep. Given an accurate estimator for Q functions, we can perform planning with standard dynamic programming (e.g. Munos & Szepesvári, 2008). However, dynamic programming in-

volves an $\arg\max$ operation, which is only tractable when $|\mathcal{A}| < \infty$. To deal with continuous actions, we leverage popular policy gradient methods like SAC (Haarnoja et al., 2018), with the critic parameterized by μLV-Rep.

To improve the exploration, we can leverage the idea of Uehara et al. (2021); Ren et al. (2023b) and add an additional ellipsoid bonus to implement optimism in the face of uncertainty. Specifically, if we use random feature quadrature, we can compute such a bonus via:

$$\hat{\psi}_{h,k}(x_h, a_h) = [\varphi(z_i; \xi_i)]_{i\in[m]},$$
$$\text{where } \{z_i\}_{i\in[m]} \sim \hat{p}_{k,h}(z|x_h, a_h), \{\xi_i\}_{i\in[m]} \sim P(\xi),$$
$$\hat{b}_{k,h}(s, a) = \alpha_k \hat{\psi}_{h,k}(x_h, a_h)\hat{\Sigma}^{-1}_{k,h}\hat{\psi}_{h,k}(x_h, a_h),$$

where $\hat{\Sigma}_{k,h} = \sum_{\mathcal{D}_{k,h}} \hat{\psi}_{k,h}(x_{h,i}, a_{h,i})\hat{\psi}_{k,h}(x_{h,i}, a_{h,i})^\top + \lambda I$, and α_k, λ are user-specified constants. Similarly, the bonus can be used to implement pessimism in the face of uncertainty for offline settings, which we defer to Appendix D.

5. Theoretical Analysis

In this section, we provide a formal sample complexity analysis of the proposed algorithm. We start from the following assumptions, that are commonly used in the literature (e.g. Agarwal et al., 2020; Uehara et al., 2021; Ren et al., 2023b).

Assumption 1 (Finite Candidate Class with Realizability). *$|\mathcal{M}| < \infty$ and $\{(p^*_h(z|x_h, a_h), p^*_h(o_{h+1}|z))\}_{h\in[H]} \in \mathcal{M}$. Meanwhile, for all $(p_h(z|x_h, a_h), p(o_{h+1}|z)) \in \mathcal{M}$, $p_h(z|x_h, a_h, o_{h+1}) \in \mathcal{Q}$.*

Assumption 2 (Normalization Conditions). *$\forall \mathcal{P} \in \mathcal{M}, (x_h, a_h) \in \mathcal{X} \times \mathcal{A}, \|p_h(\cdot|x_h, a_h)\|_{\mathcal{H}_K} \leqslant 1$ for some kernel K. Furthermore, $\forall g : \mathcal{X} \to \mathbb{R}$ such that $\|g\|_\infty \leqslant 1$, we have $\left\|\int_{\mathcal{X}} p(x_{h+L}|\cdot)g(x_{h+L})dx_{h+L}\right\|_{\mathcal{H}_K} \leqslant C$.*

Then, we have the following sample complexity bound for μLV-Rep, with detailed proofs given in Appendix F.

Theorem 3 (PAC Guarantee, Informal version of Theorem 13). *Assume the kernel K satisfies the regularity conditions in Appendix F.1. If we properly choose the exploration bonus $\hat{b}_k(x, a)$, we can obtain an ε-optimal policy with probability at least $1 - \delta$ after we interact with the environments for $N = \text{poly}\left(C, H, |\mathcal{A}|^L, L, \varepsilon, \log(|\mathcal{M}|/\delta)\right)$ episodes.*

6. Related Work

Representation has been previously considered in partially observable reinforcement learning, but for different purposes. Vision-based representations (Yarats et al., 2020; Seo et al., 2023) have been designed to extract compact features from raw pixel observations. We emphasize that this type of observation feature does not explicitly capture dynamics properties, and is essentially orthogonal to (but naturally compatible with) the proposed representation. Many dynamic-aware representation methods have been developed, such as bi-simulation (Ferns et al., 2004; Gelada et al., 2019; Zhang et al., 2020), successor features (Dayan, 1993; Barreto et al., 2017; Kulkarni et al., 2016), spectral representation (Mahadevan & Maggioni, 2007; Wu et al., 2018; Duan et al., 2019), and contrastive representation (Oord et al., 2018; Nachum & Yang, 2021; Yang et al., 2021). Our proposed representation for POMDPs has been inspired by recent progress (Jin et al., 2020b; Yang & Wang, 2020; Agarwal et al., 2020; Uehara et al., 2022) in revealing low-rank structure in the transition kernel of MDPs, and inducing effective linear representations for the state-action value function for an arbitrary policy. These prior discoveries have led to a series of practical and provable RL algorithms in the MDP setting, achieving a delicate balance between learning, planning and exploration (Ren et al., 2022; Zhang et al., 2022; Ren et al., 2023c;b). Although these algorithms demonstrate theoretical and empirical benefits, they rely on the Markovian assumption, hence are not applicable to the POMDP setting we consider here.

There have been several attempts to exploit the low-rank representation in POMDPs to reduce statistical complexity (Efroni et al., 2022). Azizzadenesheli et al. (2016) and Guo et al. (2016) exploit spectral learning for model estimation without exploration; Jin et al. (2020a) explores within an spectral estimation set; Uehara et al. (2022) builds upon the Bellman error ball; Zhan et al. (2022); Liu et al. (2022) consider the MLE confidence set for low-rank structured models; and (Huang et al., 2023) construct a UCB-type algorithm upon the MLE ball of PSR. However, these algorithms rely on intractable oracles for planning, and fewer works consider exploiting low-rank structure to achieve computationally tractable planning. One exception is (Zhang et al., 2023; Guo et al., 2023), which still includes intractable operations, *i.e.*, infinite-dimensional operations or integrals.

7. Experimental Evaluation

We evaluate the proposed method on Meta-world (Yu et al., 2019), which is an open-source simulated benchmark consisting of 50 distinct robotic manipulation tasks with visual observations. We also provide experiment results on partial observable control problems constructed based on OpenAI gym MuJoCo (Todorov et al., 2012) in Appendix H.2. Comparison on classic POMDP benchmarks, such as the Rock-Sample problem, is worth further study. (Smith & Simmons, 2012; Kurniawati et al., 2009).

We notice that directly acquiring a robust control representation through predicting future visual observations can be challenging due to the redundancy of information in images for effective decision-making. As a result, a more advantageous approach involves obtaining an image representation first and subsequently learning a latent representation based on this initial image representation. In particular, we employ visual observations with dimensions of $64 \times 64 \times 3$ and apply a variational autoendocer (VAE) (Kingma & Welling, 2013) to learn a representation of these visual observations. The VAE is trained during the online learning procedure.

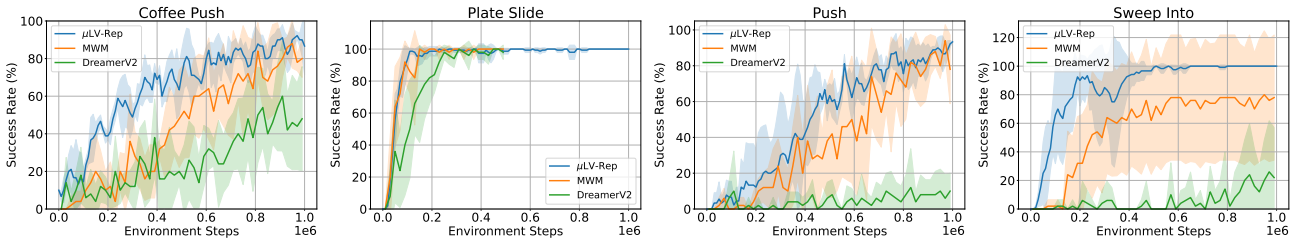


Figure 1. Learning curves on visual robotic manipulation tasks from Meta-world measured by success rate. Our method shows better or comparable sample efficiency compared to baseline methods. Learning curves on all 50 tasks are reported in Appendix H.

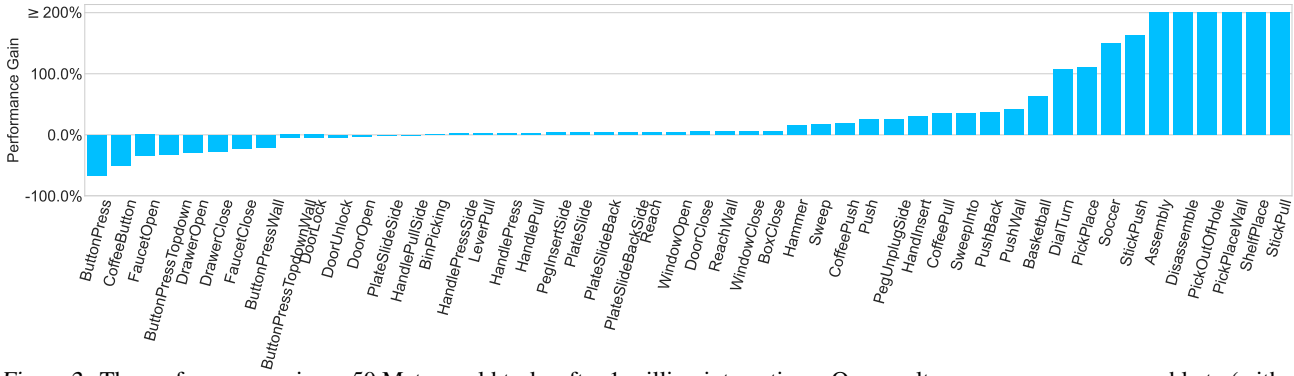


Figure 2. The performance gain on 50 Meta-world tasks after 1 million interactions. Our results surpass or are comparable to (with a difference of less than or equal to 10%) the best baselines on 41 out of the 50 tasks.

This produces compact vector representations for the images, which are then forwarded as input to the representation learning method. As detailed in Section 4.2, we learn the latent representations by making predictions about future outcomes using a history of length L. To achieve this, we employ a continuous latent variable model similar to (Ren et al., 2023b; Hafner et al., 2020), approximating distributions with Gaussians parameterized by their mean and variance. We then apply Soft Actor-Critic (SAC) as the planner (Haarnoja et al., 2018), which takes the learned representation as input of the critic. We apply $L = 3$ across all domains (with an ablation study provided in Appendix H.1). More implementation details, including network architectures and hyper-parameters, are provided in Appendix H.

We consider two baseline methods, DreamerV2 (Hafner et al., 2021) and the Masked World Model (MWM) (Seo et al., 2023). DreamerV2 is designed to acquire latent representations, which are subsequently input into a recurrent neural network for environment modeling. MWM utilizes an autoencoder equipped with convolutional layers and vision transformers (ViT) rather than reconstructing visual observations. This autoencoder is employed to reconstruct pixels based on masked convolutional features, allowing MWM to learn a latent dynamics model that operates on the representations derived from the autoencoder.

Figure 1 presents the learning curves of all algorithms as measured on the success rate for 1M environment steps, averaged over 5 random seeds. We observe that μLV-Rep exhibits superior performance compared to both DreamerV2 and MWM, demonstrating faster convergence and achieving higher final performance across all four tasks reported. We also provide the learning curves for all 50 Meta-world tasks in Appendix H.2. It is noteworthy that while MWM achieves comparable sample efficiency with μLV-Rep in certain tasks, it incurs higher computational costs and longer running times due to the incorporation of ViT network in its model. In our experimental configurations, μLV-Rep demonstrates a training speed of 21.3 steps per second, outperforming MWM, which achieves a lower training speed of 8.1 steps per second. This highlights the computational efficiency of the proposed method. Figure 5 illustrates the final performance of μLV-Rep across 50 Meta-world tasks (blue bars), comparing it with the best performance achieved by DreamerV2 and MWM (gray bars). We observe that μLV-Rep achieves better than 90% success on 33 tasks. It also performs better or equivalent to (within 10% difference) the best of DreamerV2 and MWM on 41 tasks.

8. Conclusion

In this paper, we aimed to develop a practical RL algorithm for structured POMDPs that obtained efficiency in terms of both statistical and computational complexity. We revealed some of the challenges in computationally exploiting the low-rank structure of a POMDP, then derived a linear representation for the Q^π-function, which automatically implies a practical learning method, with tractable planning and exploration, as in μLV-Rep. We theoretically analyzed the sub-optimality of the proposed μLV-Rep, and empirically demonstrated its advantages on several benchmarks.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Agarwal, A., Kakade, S., Krishnamurthy, A., and Sun, W. Flambe: Structural complexity and representation learning of low rank mdps. *Advances in neural information processing systems*, 33:20095–20107, 2020.

Aronszajn, N. Theory of reproducing kernels. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

Åström, K. J. Optimal control of markov processes with incomplete state information. *Journal of mathematical analysis and applications*, 10(1):174–205, 1965.

Azizzadenesheli, K., Lazaric, A., and Anandkumar, A. Reinforcement learning of pomdps using spectral methods. In *COLT*, 2016.

Bakker, B. Reinforcement learning with long short-term memory. *NeurIPS*, 2001.

Barreto, A., Dabney, W., Munos, R., Hunt, J. J., Schaul, T., van Hasselt, H. P., and Silver, D. Successor features for transfer in reinforcement learning. *NeurIPS*, 2017.

Barth-Maron, G., Hoffman, M. W., Budden, D., Dabney, W., Horgan, D., Tb, D., Muldal, A., Heess, N., and Lillicrap, T. Distributed distributional deterministic policy gradients. *arXiv preprint arXiv:1804.08617*, 2018.

Berner, C., Brockman, G., Chan, B., Cheung, V., Dębiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

Chen, M., Frazzoli, E., Hsu, D., and Lee, W. S. Pomdp-lite for robust robot planning under uncertainty. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5427–5433. IEEE, 2016.

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

Deisenroth, M. P. and Peters, J. Solving nonlinear continuous state-action-observation pomdps for mechanical systems with gaussian noise. In *European Workshop on Reinforcement Learning*, 2012.

Du, S., Krishnamurthy, A., Jiang, N., Agarwal, A., Dudik, M., and Langford, J. Provably efficient rl with rich observations via latent state decoding. In *International Conference on Machine Learning*, pp. 1665–1674. PMLR, 2019.

Duan, Y., Ke, T., and Wang, M. State aggregation learning from markov transition data. *NeurIPS*, 2019.

Efroni, Y., Jin, C., Krishnamurthy, A., and Miryoosefi, S. Provable reinforcement learning with a short-term memory. In *International Conference on Machine Learning*, pp. 5832–5850. PMLR, 2022.

Even-Dar, E., Kakade, S. M., and Mansour, Y. The value of observation for monitoring dynamic systems. In *IJCAI*, pp. 2474–2479, 2007.

Ferns, N., Panangaden, P., and Precup, D. Metrics for finite markov decision processes. In *UAI*, 2004.

Gangwani, T., Lehman, J., Liu, Q., and Peng, J. Learning belief representations for imitation learning in pomdps. In *UAI*, 2020.

Gelada, C., Kumar, S., Buckman, J., Nachum, O., and Bellemare, M. G. Deepmdp: Learning continuous latent space models for representation learning. In *ICML*, 2019.

Golowich, N., Moitra, A., and Rohatgi, D. Learning in observable pomdps, without computationally intractable oracles. *Advances in Neural Information Processing Systems*, 35:1458–1473, 2022.

Gregor, K., Jimenez Rezende, D., Besse, F., Wu, Y., Merzic, H., and van den Oord, A. Shaping belief states with generative environment models for rl. *NeurIPS*, 2019.

Guo, J., Li, Z., Wang, H., Wang, M., Yang, Z., and Zhang, X. Provably efficient representation learning with tractable planning in low-rank pomdp. *arXiv preprint arXiv:2306.12356*, 2023.

Guo, Z. D., Doroudi, S., and Brunskill, E. A pac rl algorithm for episodic pomdps. In *Artificial Intelligence and Statistics*, pp. 510–518. PMLR, 2016.

Guo, Z. D., Azar, M. G., Piot, B., Pires, B. A., and Munos, R. Neural predictive belief representations. *arXiv preprint arXiv:1811.06407*, 2018.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Hafner, D., Lillicrap, T. P., Norouzi, M., and Ba, J. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=0oabwyZbOu.

Hausknecht, M. and Stone, P. Deep recurrent q-learning for partially observable mdps. In *AAAI fall symposium series*, 2015.

Hauskrecht, M. and Fraser, H. Planning treatment of ischemic heart disease with partially observable markov decision processes. *Artificial intelligence in medicine*, 18 (3):221–244, 2000.

Heess, N., Hunt, J. J., Lillicrap, T. P., and Silver, D. Memory-based control with recurrent neural networks. *arXiv:1512.04455*, 2015.

Huang, R., Liang, Y., and Yang, J. Provably efficient ucb-type algorithms for learning predictive state representations. *arXiv preprint arXiv:2307.00405*, 2023.

Igl, M., Zintgraf, L., Le, T. A., Wood, F., and Whiteson, S. Deep variational reinforcement learning for pomdps. In *ICML*, 2018.

Jiang, H., Dai, B., Yang, M., Zhao, T., and Wei, W. Towards automatic evaluation of dialog systems: A model-free off-policy evaluation approach. *arXiv preprint arXiv:2102.10242*, 2021.

Jin, C., Kakade, S., Krishnamurthy, A., and Liu, Q. Sample-efficient reinforcement learning of undercomplete pomdps. *Advances in Neural Information Processing Systems*, 33:18530–18539, 2020a.

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. Provably efficient reinforcement learning with linear function approximation. In *COLT*, 2020b.

Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Kaufmann, E., Bauersfeld, L., Loquercio, A., Müller, M., Koltun, V., and Scaramuzza, D. Champion-level drone racing using deep reinforcement learning. *Nature*, 620 (7976):982–987, 2023.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Kulkarni, T. D., Saeedi, A., Gautam, S., and Gershman, S. J. Deep successor reinforcement learning. *arXiv:1606.02396*, 2016.

Kurniawati, H., Hsu, D., and Lee, W. S. Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces. 2009.

Lee, A. X., Nagabandi, A., Abbeel, P., and Levine, S. Stochastic latent actor-critic: Deep reinforcement learning with a latent variable model. *NeurIPS*, 2020.

Levine, S., Finn, C., Darrell, T., and Abbeel, P. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

Littman, M. and Sutton, R. S. Predictive representations of state. *Advances in neural information processing systems*, 14, 2001.

Liu, Q., Chung, A., Szepesvári, C., and Jin, C. When is partially observable reinforcement learning not scary? In *Conference on Learning Theory*, pp. 5175–5220. PMLR, 2022.

Liu, Q., Netrapalli, P., Szepesvari, C., and Jin, C. Optimistic mle: A generic model-based algorithm for partially observable sequential decision making. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pp. 363–376, 2023.

Mahadevan, S. and Maggioni, M. Proto-value functions: A laplacian framework for learning representation and control in markov decision processes. *JMLR*, 8(10), 2007.

Meng, L., Gorbet, R., and Kulić, D. Memory-based deep reinforcement learning for pomdps. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5619–5626. IEEE, 2021.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Munos, R. and Szepesvári, C. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9 (5), 2008.

Nachum, O. and Dai, B. Reinforcement learning via fenchel-rockafellar duality. *arXiv preprint arXiv:2001.01866*, 2020.

Nachum, O. and Yang, M. Provable representation learning for imitation with contrastive fourier features. *NeurIPS*, 2021.

Ni, T., Eysenbach, B., and Salakhutdinov, R. Recurrent model-free rl is a strong baseline for many pomdps. *arXiv:2110.05038*, 2021.

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

Papadimitriou, C. H. and Tsitsiklis, J. N. The complexity of markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.

Paulsen, V. I. and Raghupathi, M. *An introduction to the theory of reproducing kernel Hilbert spaces*, volume 152. Cambridge university press, 2016.

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

Ren, T., Zhang, T., Szepesvári, C., and Dai, B. A free lunch from the noise: Provable and practical exploration for representation learning. In *Uncertainty in Artificial Intelligence*, pp. 1686–1696. PMLR, 2022.

Ren, T., Ren, Z., Li, N., and Dai, B. Stochastic nonlinear control via finite-dimensional spectral dynamic embedding. *arXiv preprint arXiv:2304.03907*, 2023a.

Ren, T., Xiao, C., Zhang, T., Li, N., Wang, Z., Sanghavi, S., Schuurmans, D., and Dai, B. Latent variable representation for reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023b.

Ren, T., Zhang, T., Lee, L., Gonzalez, J. E., Schuurmans, D., and Dai, B. Spectral decomposition representation for reinforcement learning. *ICLR*, 2023c.

Riesz, F. and Nagy, B. S. *Functional analysis*. Courier Corporation, 2012.

Ross, S., Pineau, J., Paquet, S., and Chaib-Draa, B. Online planning algorithms for pomdps. *Journal of Artificial Intelligence Research*, 32:663–704, 2008.

Roy, N. and Gordon, G. J. Exponential family pca for belief compression in pomdps. *Advances in Neural Information Processing Systems*, 15, 2002.

Schmidhuber, J. Reinforcement learning in markovian and non-markovian environments. *NeurIPS*, 3, 1990.

Seo, Y., Hafner, D., Liu, H., Liu, F., James, S., Lee, K., and Abbeel, P. Masked world models for visual control. In *Conference on Robot Learning*, pp. 1332–1344. PMLR, 2023.

Simchowitz, M., Tosh, C., Krishnamurthy, A., Hsu, D. J., Lykouris, T., Dudik, M., and Schapire, R. E. Bayesian decision-making under misspecified priors with applications to meta-learning. *Advances in Neural Information Processing Systems*, 34:26382–26394, 2021.

Smallwood, R. D. and Sondik, E. J. The optimal control of partially observable markov processes over a finite horizon. *Operations research*, 21(5):1071–1088, 1973.

Smith, T. and Simmons, R. Heuristic search value iteration for pomdps. *arXiv preprint arXiv:1207.4166*, 2012.

Steinwart, I. and Scovel, C. Mercer's theorem on general domains: On the interaction between measures, kernels, and rkhss. *Constructive Approximation*, 35(3):363–417, 2012.

Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., Casas, D. d. L., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A., et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Uehara, M., Zhang, X., and Sun, W. Representation learning for online and offline rl in low-rank mdps. *arXiv preprint arXiv:2110.04652*, 2021.

Uehara, M., Sekhari, A., Lee, J. D., Kallus, N., and Sun, W. Provably efficient reinforcement learning in partially observable dynamical systems. *Advances in Neural Information Processing Systems*, 35:578–592, 2022.

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

Weigand, S., Klink, P., Peters, J., and Pajarinen, J. Reinforcement learning using guided observability. *arXiv:2104.10986*, 2021.

Wierstra, D., Foerster, A., Peters, J., and Schmidhuber, J. Solving deep memory pomdps with recurrent policy gradients. In *ICANN*, 2007.

Wu, Y., Tucker, G., and Nachum, O. The laplacian in rl: Learning representations with efficient approximations. *arXiv:1810.04586*, 2018.

Yang, L. and Wang, M. Reinforcement learning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*, pp. 10746–10756. PMLR, 2020.

Yang, M., Levine, S., and Nachum, O. Trail: Near-optimal imitation learning with suboptimal data. *arXiv:2110.14770*, 2021.

Yarats, D., Kostrikov, I., and Fergus, R. Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In *International conference on learning representations*, 2020.

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

Yarats, D., Fergus, R., Lazaric, A., and Pinto, L. Mastering visual continuous control: Improved data-augmented reinforcement learning. In *International Conference on Learning Representations*, 2022.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2019. URL https://arxiv.org/abs/1910.10897.

Zhan, W., Uehara, M., Sun, W., and Lee, J. D. Pac reinforcement learning for predictive state representations. *arXiv preprint arXiv:2207.05738*, 2022.

Zhang, A., McAllister, R., Calandra, R., Gal, Y., and Levine, S. Learning invariant representations for reinforcement learning without reconstruction. *arXiv:2006.10742*, 2020.

Zhang, M., Vikram, S., Smith, L., Abbeel, P., Johnson, M., and Levine, S. Solar: Deep structured representations for model-based reinforcement learning. In *ICML*, 2019.

Zhang, T., Ren, T., Yang, M., Gonzalez, J., Schuurmans, D., and Dai, B. Making linear mdps practical via contrastive representation learning. In *ICML*, 2022.

Zhang, T., Ren, T., Xiao, C., Xiao, W., Gonzalez, J., Schuurmans, D., and Dai, B. Energy-based predictive representations for partially observed reinforcement learning. In *The 39th Conference on Uncertainty in Artificial Intelligence*, 2023. URL https://openreview.net/forum?id=Z1BobPmCTy.

Zhu, P., Li, X., Poupart, P., and Miao, G. On improving deep reinforcement learning for pomdps. *arXiv preprint arXiv:1704.07978*, 2017.

A. More Related Work

Partially Observable RL. The majority of existing practical RL algorithms for partially observable settings can also be categorized into model-based *vs*. model-free.

The model-based algorithms (Kaelbling et al., 1998) for partially observed scenarios are naturally derived based on the definition of POMDPs, where both the emission and transition models are learned from data. The planning procedure for optimal policy is conducted over the posterior of latent state, *i.e.*, beliefs, which is approximately inferred based on learned dynamics and emission model. With different model parametrizations, (ranging from Gaussian processes to deep models), and different planning methods, a family of algorithms has been proposed (Deisenroth & Peters, 2012; Igl et al., 2018; Gregor et al., 2019; Zhang et al., 2019; Lee et al., 2020; Hafner et al., 2021). However, due to the compounding errors from **i)**, mismatch in model parametrization, **ii)**, inaccurate beliefs calculation, **iii)**, approximation in planning over nonlinear dynamics, and **iv)**, neglecting of exploration, such methods might suffer from sub-optimal performances in practice.

As we discussed in Section 1, the memory-based policy and value function have been exploited to extend the MDP-based model-free RL algorithms to handle the non-Markovian dependency induced by partial observations. For example, the value-based algorithms introduces memory-based neural networks to Bellman recursion, including temporal difference learning with explicit concatenation of 4 consecutive frames as input (Mnih et al., 2013) or recurrent neural networks for longer windows (Bakker, 2001; Hausknecht & Stone, 2015; Zhu et al., 2017), and DICE (Nachum & Dai, 2020) with features extracted from transformer (Jiang et al., 2021); the policy gradient-based algorithms have been extended to partially observable setting by introducing recurrent neural network for policy parametrization (Schmidhuber, 1990; Wierstra et al., 2007; Heess et al., 2015; Ni et al., 2021). The actor-critic approaches exploits memory-based value and policy together (Ni et al., 2021; Meng et al., 2021). Despite their simplicity in the algorithm extension, these algorithms demonstrate potentials in real-world applications. However, the it has been observed that the sample complexity for purely model-free RL with partial observations is very high (Mnih et al., 2013; Barth-Maron et al., 2018; Yarats et al., 2021), and the exploration remains difficult, and thus, largely neglected.

B. Observability Approximation

Although the proposed μLV-Rep is designed based on the L-step decodability in POMDPs, Golowich et al. (2022) shows that the γ-observable POMDPs can be ϵ-approximated with a $L = \tilde{O}\left(\gamma^{-4}\log\left(|S|/\epsilon\right)\right)$-step decodable POMDP. By exploiting the low-rank structure in the latent dynamics, this result has been extend with function approximator (Uehara et al., 2022). Specifically,

Theorem 4 (Proprosition 7 (Guo et al., 2023), Lemma 12 (Uehara et al., 2022))**.** *Given a γ-observable POMDP with d-rank latent transition, there exists an L-step decodable POMDP $\mathcal{M}$ with $L = \tilde{O}\left(\gamma^{-4}\log\left(d/\epsilon\right)\right)$, $\forall \epsilon > 0$, such that*

$$\mathbb{E}_{a_{1:h}, o_{2:h}\sim\pi}\left[\left\|\mathbb{P}_h\left(o_{h+1}|o_{1:h}, a_{1:h}\right) - \mathbb{P}_h^{\mathcal{M}}\left(o_{h+1}|x_h, a_h\right)\right\|_1\right] \leqslant \epsilon. \tag{15}$$

where $\pi_h \in \Delta\left(\prod_{h=1}^{H}\mathcal{A}^{\mathscr{H}_h}\right)$ with $\mathscr{H}_h := \mathcal{A}^{h-1}\times\mathcal{O}^h$, is mapping the whole history to a distribution of action.

With this understanding, the proposed μLV-Rep can be directly applied for γ-observable POMDPs, while still maintains theoretical guarantees. Due to the space limitation, please refer to (Uehara et al., 2022; Guo et al., 2023) for the details of the proofs.

C. Moment Matching Policy

We provide a formal definition of the moment matching policy here.

Definition 5 (Moment Matching Policy (Efroni et al., 2022))**.** *With the L-decodability assumption, for $h \in [H]$, $h' \in [h-L+1, h]$ and $l = h'-h+L-1$, we can define the moment matching policy $\nu^{\pi,h} = \{\nu_{h'}^{h,\pi} : \mathcal{S}^l\times\mathcal{O}^l\times\mathcal{A}^{l-1} \to \Delta(\mathcal{A})\}_{h'=h-L+1}^{h}$ introduced by Efroni et al. (2022) , such that*

$$\begin{aligned}&\nu_{h'}^{\pi,h}(a_{h'}|(s_{h-L+1:h'}, o_{h-L+1:h'}, a_{h-L+1:h'-1}))\\ &\qquad := \mathbb{E}_{\pi}^{\mathcal{P}}[\pi_{h'}(a_{h'}|x_{h'})|(s_{h-L+1:h'}, o_{h-L+1:h'}, a_{h-L+1:h'-1})], \quad \forall h' \leqslant h-1,\end{aligned}$$

and $\nu_h^{\pi,h} = \pi_h$. We further define $\tilde{\pi}^h$, which takes first $h-L$ actions from π and the remaining L actions from $\nu^{\pi,h}$.

The main motivation to define such moment matching policy is that, we want to define a policy that is conditionally independent from the past history for theoretical justification while indistinguishable from the history dependent policy

Algorithm 2 Offline Learning for L-step decodable POMDPs with Latent Variable Representation

1: **Input:** Model Class $\mathcal{M} = \{\{(p_h(z|x_h, a_h), p_h(o_{h+1}|z)\}_{h\in[H]}\}$, Variational Distribution Class $\mathcal{Q} = \{\{q_h(z|x_h, a_h, o_{h+1})\}_{h\in[H]}\}$, Offline Dataset $\{\mathcal{D}_h\}_{h=1}^H$
2: Learn the latent variable model $\hat{p}(z|x_h, a_h)$ with $\mathcal{D}_h$ via maximizing the ELBO, and obtain the learned model $\widehat{\mathcal{P}} = \{(\hat{p}_h(z|x_h, a_h), \hat{p}_h(o_{h+1}|z))\}_{h\in[H]}$.
3: Set the exploitation penalty $\hat{b}_h(s, a)$ with $\mathcal{D}_k$.
4: Learn the policy $\hat{\pi}^* = \arg\max_\pi V^\pi_{\widehat{\mathcal{P}}, r-\hat{b}_k}$.
5: **Return** $\hat{\pi}^*$.

to match the practical algorithm. By Lemma B.2 in Efroni et al. (2022), under the L-decodability assumption, for a fixed $h \in [H]$, we have $d_h^{\mathcal{P},\pi}(x_h) = d_h^{\mathcal{P},\tilde{\pi}_h}(x_h)$, for all L-step policy π and $x_h \in \mathcal{X}_h$. As $\nu_h^{\pi,h} = \pi_h$. we have $d_h^{\mathcal{P},\pi}(x_h, a_h) = d_h^{\mathcal{P},\tilde{\pi}^h}(x_h, a_h)$, and hence $\mathbb{E}_\pi^{\mathcal{P}}(x_h, a_h) = \mathbb{E}_{\tilde{\pi}^h}^{\mathcal{P}}(x_h, a_h)$. This enables the factorization in (10) without the dependency of the overlap observation trajectory.

D. Pessimism in the Offline Setting

Similar to Uehara et al. (2021); Ren et al. (2023b), the proposed algorithm can be directly extended to the offline setting by converting the optimism into the pessimism. Specifically, we can learn the latent variable model, set the penalty with the data and perform planning with the penalized reward. The whole algorithm is shown in Algorithm 2. Following the identical proof strategy from Uehara et al. (2021); Ren et al. (2023b), we can obtain a similar sub-optimal gap guarantee for $\hat{\pi}^*$.

E. Technical Background

In this section, we revisit several core concepts of the kernel and the reproducing kernel Hilbert space (RKHS) that will be used in the theoretical analysis. For a complete introduction, we refer the reader to Ren et al. (2023b).

Definition 6 (Kernel and Reproducing Kernel Hilbert Space (RKHS) (Aronszajn, 1950; Paulsen & Raghupathi, 2016))**.** *The function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a kernel on $\mathcal{X}$ if there exists a Hilbert space $\mathcal{H}$ and a mapping $\phi : \mathcal{X} \to \mathcal{H}$ (termed as a feature map), such that $\forall x, x' \in \mathcal{X}$, $k(x, x') = \langle \phi(x), \phi(x')\rangle_{\mathcal{H}}$. The kernel k is said to be positive semi-definite if $\forall n \geqslant 1, \{a_i\}_{i\in[n]} \subset \mathbb{R}$ and mutually distinct $\{x_i\}_{i\in[n]}$, we have*

$$\sum_{i\in[n]}\sum_{j\in[n]} a_i a_j k(x_i, x_j) \geqslant 0.$$

The kernel k is said to be positive definite if the inequality is strict (which means we can replace $\geqslant$ with $>$).

With a given kernel k, we can define the Hilbert space $\mathcal{H}_k$ consists of $\mathbb{R}$-valued function on $\mathcal{X}$ as a reproducing kernel Hilbert space associated with k if both of the following conditions hold:

- *$\forall x \in \mathcal{X}$, $k(x, \cdot) \in \mathcal{H}_k$.*
- *Reproducing Property: $\forall x \in \mathcal{X}, f \in \mathcal{H}_k, f(x) = \langle f, k(x, \cdot)\rangle_{\mathcal{H}_k}$.*

The RKHS norm of $f \in \mathcal{H}_k$ is induced by the inner product, i.e. $\|f\|_{\mathcal{H}_k} := \sqrt{\langle f, f\rangle_{\mathcal{H}_k}}$.

Theorem 7 (Mercer's Theorem (Riesz & Nagy, 2012; Steinwart & Scovel, 2012))**.** *Let k be a continuous positive definite kernel defined on $\mathcal{X} \times \mathcal{X}$. There exists at most countable $\{\mu_i\}_{i\in I}$ such that $\mu_1 \geqslant \mu_2 \geqslant \cdots > 0$ and a set of orthonormal basis $\{e_i\}_{i\in I}$ on $L_2(\mu)$ where μ is a Borel measure on $\mathcal{X}$, such that*

$$\forall x, x' \in \mathcal{X}, \quad k(x, x') = \sum_{i\in I} \mu_i e_i(x) e_i(x'),$$

where the convergence is absolute and uniform.

Definition 8 (Random Feature)**.** *The kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ has a random feature representation if there exists a function $\psi : \mathcal{X} \times \Xi \to \mathbb{R}$ and a probability measure P over Ξ such that*

$$k(x, x') = \int_\Xi \psi(x; \xi)\psi(x'; \xi) dP(\xi).$$

Remark (random feature quadrature): We here justify the random feature quadrature (Ren et al., 2023b) for completeness.

We can represent Q_h^π as an expectation,

$$Q_h^\pi(x_h, a_h) = \langle p(z|x_h), w_h^\pi(z)\rangle = \mathbb{E}_{p(z|x_h)}\left[w_h^\pi(z)\right]_{L_2(\mu)}$$

Under the assumption that $w_h^\pi(\cdot) \in \mathcal{H}_k$, where $\mathcal{H}_k$ denoting some RKHS with some kernel $k(\cdot,\cdot)$. When $k(\cdot,\cdot)$ can be represented through random feature, *i.e.*,

$$k(x, y) = \mathbb{E}_{P(\xi)}\left[\psi(x;\xi)\,\psi(y;\xi)\right],$$

the $w_h^\pi(z)$ admits a representation as

$$w_h^\pi(z) = \mathbb{E}_{P(\xi)}\left[\tilde{w}_h^\pi(\xi)\,\psi(z;\xi)\right].$$

Therefore, we plug this random feature representation of $w_h^\pi(z)$ to $Q_h^\pi(x_h, a_h)$, we obtain

$$Q_h^\pi(x_h, a_h) = \mathbb{E}_{p(z|x_h),P(\xi)}\left[\tilde{w}_h^\pi(\xi)\psi(z;\xi)\right]. \tag{16}$$

Applying Monte-Carlo approximation to (16), we obtain the random feature quadrature.

F. Theoretical Analysis

F.1. Technical Conditions

We adopt the following assumptions for the reproducing kernel, which have been used in Ren et al. (2023b) for the MDP setting.

Assumption 3 (Regularity Conditions)**.** *$\mathcal{Z}$ is a compact metric space with respect to the Lebesgue measure ν when $\mathcal{Z}$ is continuous. Furthermore, $\int_{\mathcal{Z}} k(z,z)d\nu \leqslant 1$.*

Assumption 4 (Eigendecay Conditions)**.** *Assume $\{\nu_i\}_{i\in I}$ defined in Theorem 7 satisfies one of the following conditions:*

- *β-finite spectrum: for some positive integer β, we have $\nu_i = 0$, $\forall i > \beta$.*
- *β-polynomial decay: $\nu_i \leqslant C_0 i^{-\beta}$ with absolute constant C_0 and $\beta > 1$.*
- *β-exponential decay: $\nu_i \leqslant C_1 \exp(-C_2 i^\beta)$, with absolute constants C_1, C_2 and $\beta > 0$.*

We will use C_{poly} to denote constants in the analysis of β-polynomial decay that only depends on C_0 and β, and C_{exp} to denote constants in the analysis of β-exponential decay that only depends on C_1, C_2 and β, to simplify the dependency of the constant terms. Both of C_{poly} and C_{exp} can be varied step by step.

F.2. Formal Proof

Before we proceed, we first define

$$\rho_{k,h} = \frac{1}{k}\sum_{i\in[K]} d_{\mathcal{P},h}^{\pi_k},$$

and $\circ^L\mathcal{U}(\mathcal{A})$ means uniformly taking actions in the consecutive L steps.

Lemma 9 (L-step back inequality for the true model)**.** *Given a set of functions $[g_h]_{h\in[H]}$, where $g_h : \mathcal{X}\times\mathcal{A}\to\mathbb{R}$, $\|g_h\|_\infty \leqslant B$, $\forall h\in[H]$, we have that $\forall\pi$,*

$$\sum_{h\in[H]} \mathbb{E}_\pi^{\mathcal{P}}[g(x_h,a_h)] \leqslant \sum_{h\in[H]} \mathbb{E}^{\mathcal{P}}_{(x_{h-L},a_{h-L})\sim d^\pi_{\mathcal{P},h-L}}\left[\|p^*(\cdot|x_{h-L},a_{h-L})\|_{L_2(\mu),\Sigma^{-1}_{\rho_{k,h-L},p^*}}\right]$$
$$\cdot\sqrt{k|\mathcal{A}|^L\cdot\mathbb{E}_{(\tilde{x}_h,\tilde{a}_h)\sim\rho_{k,h-L}\circ^L\mathcal{U}(\mathcal{A})}\left[g(\tilde{x}_h,\tilde{a}_h)^2\right] + \lambda B^2 C}$$

Proof. The proof can be adapted from the proof of Lemma 6 in Ren et al. (2023b), and we include it for the completeness. Recall the moment matching policy ν^π Since $\nu^{\pi,h}$ does not depend on (x_{h-L}, a_{h-L}), we can make the following

decomposition:

$$
\begin{aligned}
&\mathbb{E}^{\mathcal{P}}_{\tilde{\pi}^h}(x_h, a_h) \\
=&\mathbb{E}^{\mathcal{P}}_{(x_{h-L}, a_{h-L})\sim\pi}\left[\int_{s_{h-L+1}} \langle p^*(\cdot|x_{h-L}, a_{h-L}), p^*(s_{h-L+1}|\cdot)\rangle_{L_2(\mu)} \cdot \mathbb{E}^{\mathcal{P}}_{a_{h-L+1:h}\sim\nu^{\pi,h}}[g(x_h, a_h)|s_{h-L+1}]ds_{h-L+1}\right] \\
\leqslant&\mathbb{E}^{\mathcal{P}}_{(x_{h-L}, a_{h-L})\sim\pi} \|p^*(\cdot|x_{h-L}, a_{h-L})\|_{L_2(\mu), \Sigma^{-1}_{\rho_{k,h-L}, p^*}} \\
&\cdot \left\|\int_{s_{h-L+1}} p^*(s_{h-L+1}|\cdot)\mathbb{E}[g(x_h, a_h)|s_{h-L+1}, \nu^{\pi,h}]ds_{h-L+1}\right\|_{L_2(\mu), \Sigma_{\rho_{k,h-L}, p^*}}.
\end{aligned}
$$

Direct computation shows that

$$
\begin{aligned}
&\left\|\int_{s_{h-L+1}} p^*(s_{h-L+1}|\cdot)\mathbb{E}^{\mathcal{P}}[g(x_h, a_h)|s_{h-L+1}, \nu^{\pi,h}]ds_{h-L+1}\right\|_{L_2(\mu), \Sigma_{\rho_{k,h-L}, p^*}} \\
=&k\mathbb{E}_{(\tilde{x}_{h-L}, \tilde{a}_{h-L})\sim\rho_{k,h-L}}\left[\mathbb{E}^{\mathcal{P}}_{s_{h-L+1}\sim\mathbb{P}^{\mathcal{P}}_{h-L}(\cdot|x_{h-L}, a_{h-L})}[g(x_h, a_h)|s_{h-L+1}, \nu^{\pi,h}]\right]^2 \\
&+\left\|\int_{s_{h-L+1}} p^*(s_{h-L+1}|\cdot)\cdot\mathbb{E}^{\mathcal{P}}[g(x_h, a_h)|s_{h-L+1}, \nu^{\pi,h}]ds_{h-L+1}\right\|^2_{\mathcal{H}} \\
\leqslant&k\mathbb{E}_{(\tilde{x}_{h-L}, \tilde{a}_{h-L})\sim\rho_{k,h-L}}\mathbb{E}^{\mathcal{P}}_{s_{h-L+1}\sim\mathbb{P}^{\mathcal{P}}_{h-L}(\cdot|x_{h-L}, a_{h-L}), a_{h-L+1:h}\sim\nu^{\pi,h}}[g(x_h, a_h)]^2 + \lambda B^2 C \\
\leqslant&k|\mathcal{A}|^L\mathbb{E}^{\mathcal{P}}_{(\tilde{x}_h, \tilde{a}_h)\sim\rho_{k,h-L}\circ^L\mathcal{U}(\mathcal{A})}[g(\tilde{x}_h, \tilde{a}_h)]^2 + \lambda B^2 C,
\end{aligned}
$$

which finishes the proof. □

Lemma 10 (L-step back inequality for the learned model)**.** *Assume we have a set of functions* $[g_h]_{h\in[H]}$*, where* $g_h : \mathcal{X}\times\mathcal{A}\to\mathbb{R}$*,* $\|g_h\|_\infty \leqslant B$*,* $\forall h\in[H]$*. Given Lemma 15, we have that* $\forall\pi$*,*

$$
\begin{aligned}
\sum_{h\in[H]}\mathbb{E}^{\widehat{\mathcal{P}}_k}_{\pi}[g(x_h, a_h)] \leqslant& \sum_{h\in[H]}\mathbb{E}^{\widehat{\mathcal{P}}_k}_{(x_{h-L}, a_{h-L})\sim d^{\pi}_{\widehat{\mathcal{P}}_k, h-L}}\left[\|\hat{p}(\cdot|x_{h-L}, a_{h-L})\|_{L_2(\mu), \Sigma^{-1}_{\rho_{k,h-2L}\circ^L\mathcal{U}(\mathcal{A}), \hat{p}}}\right] \\
&\cdot\sqrt{k|\mathcal{A}|^L\cdot\mathbb{E}_{(\tilde{x}_h, \tilde{a}_h)\sim\rho_{k,h-2L}\circ^{2L}\mathcal{U}(\mathcal{A})}\left[g(\tilde{x}_h, \tilde{a}_h)^2\right] + \lambda B^2 C + kL|\mathcal{A}|^{L-1}B^2\zeta_k}
\end{aligned}
$$

Proof. The proof can be adapted from the proof of Lemma 5 in Ren et al. (2023b), and we include it for the completeness. We define a similar moment matching policy and make the following decomposition:

$$
\begin{aligned}
&\mathbb{E}^{\widehat{\mathcal{P}}_k}_{\pi}(x_h, a_h) \\
=&\mathbb{E}^{\widehat{\mathcal{P}}_k}_{(x_{h-L}, a_{h-L})\sim\pi}\left[\int_{s_{h-L+1}} \langle \hat{p}(\cdot|x_{h-L}, a_{h-L}), \hat{p}(s_{h-L+1}|\cdot)\rangle_{L_2(\mu)} \cdot \mathbb{E}^{\widehat{\mathcal{P}}_k}[g(x_h, a_h)|s_{h-L+1}, \nu^{\pi,h}]ds_{h-L+1}\right] \\
\leqslant&\mathbb{E}^{\widehat{\mathcal{P}}_k}_{(x_{h-L}, a_{h-L})\sim\pi} \|\hat{p}(\cdot|x_{h-L}, a_{h-L})\|_{L_2(\mu), \Sigma^{-1}_{\rho_{k,h-2L}\circ^L\mathcal{U}(\mathcal{A}), \hat{p}}} \\
&\cdot \left\|\int_{s_{h-L+1}} \hat{p}(s_{h-L+1}|\cdot)\mathbb{E}^{\widehat{\mathcal{P}}_k}[g(x_h, a_h)|s_{h-L+1}, \nu^{\pi,h}]ds_{h-L+1}\right\|_{L_2(\mu), \Sigma_{\rho_{k,h-2L}\circ^L\mathcal{U}(\mathcal{A}), \hat{p}}}.
\end{aligned}
$$

Direct computation shows that

$$
\begin{aligned}
&\left\| \int_{s_{h-L+1}} \hat{p}(s_{h-L+1}|\cdot)\mathbb{E}^{\widehat{\mathcal{P}}_k}[g(x_h,a_h)|s_{h-L+1},\nu^{\pi,h}]ds_{h-L+1} \right\|^2_{L_2(\mu),\Sigma_{\rho_{k,h-2L}\circ^L\mathcal{U}(\mathcal{A}),\hat{p}}} \\
=& k\mathbb{E}_{(\tilde{x}_{h-L},\tilde{a}_{h-L})\sim\rho_{k,h-2L}\circ^L\mathcal{U}(\mathcal{A})}\left[\mathbb{E}_{s_{h-L+1}\sim\mathbb{P}^{\widehat{\mathcal{P}}_k}_{h-L}(\cdot|\tilde{x}_{h-L},\tilde{a}_{h-L})}\mathbb{E}^{\widehat{\mathcal{P}}_k}[g(x_h,a_h)|s_{h-L+1},\nu^{\pi,h}]\right]^2 \\
&+\left\| \int_{s_{h-L+1}} \hat{p}(s_{h-L+1}|\cdot)\mathbb{E}[g(x_h,a_h)|s_{h-L+1},\nu^{\pi,h}]ds_{h-L+1} \right\|^2_{\mathcal{H}} \\
\leqslant& k\mathbb{E}_{(\tilde{x}_{h-L},\tilde{a}_{h-L})\sim\rho_{k,h-2L}\circ^L\mathcal{U}(\mathcal{A})}\mathbb{E}^{\widehat{\mathcal{P}}_k}_{s_{h-L+1}\sim\mathbb{P}^{\widehat{\mathcal{P}}_k}_{h-L}(\cdot|\tilde{x}_{h-L},\tilde{a}_{h-L}),\nu^{\pi,h}}[g(x_h,a_h)]^2+\lambda B^2C \\
\leqslant& k|\mathcal{A}|^L\mathbb{E}_{(\tilde{x}_{h-L},\tilde{a}_{h-L})\sim\rho_{k,h-2L}\circ^L\mathcal{U}(\mathcal{A})}\mathbb{E}^{\widehat{\mathcal{P}}_k}_{a_{h-L+1:h}\sim\circ^L\mathcal{U}(\mathcal{A})}[g(x_h,a_h)]^2+\lambda B^2C \\
\leqslant& k|\mathcal{A}|^L\mathbb{E}_{(\tilde{x}_h,\tilde{a}_h)\sim\rho_{k,h-2L}\circ^{2L}\mathcal{U}(\mathcal{A})}[g(\tilde{x}_h,\tilde{a}_h)]^2+kL|\mathcal{A}|^{L-1}B^2\zeta_k+\lambda B^2C,
\end{aligned}
$$

where we use the MLE guarantee for each individual step to obtain the last inequality. This finishes the proof. □

Lemma 11 (Almost Optimism)**.** *For episode $k\in[K]$, set*

$$\hat{b}_{k,h}=\min\left\{\alpha_k\|\hat{p}_k(\cdot|x_{h-L},a_{h-L})\|_{L_2(\mu),\hat{\Sigma}^{-1}_{k,h,\hat{p}_k}},2\right\},$$

with $\alpha_k=\frac{\sqrt{5kL|\mathcal{A}|^L\zeta_k+4\lambda d}}{c}$,

$$\hat{\Sigma}_{k,h,\hat{p}_k}:L_2(\mu)\to L_2(\mu),\quad \hat{\Sigma}_{k,h,\hat{p}_k}:=\sum_{(x_{h,i},a_{h,i})\in\mathcal{D}_{k,h}}\left[\hat{p}_k(z|x_{h,i},a_{h,i})\hat{p}_k(z|x_{h,i},a_{h,i})^\top\right]+\lambda T_K^{-1}$$

where T_K is the integral operator associated with K (i.e. $T_Kf=\int f(x)K(x,\cdot)dx$) and λ is set for different eigendecay of K as follows:

- *β-finite spectrum:* $\lambda=\Theta(\beta\log K+\log(K|\mathcal{P}|/\delta))$
- *β-polynomial decay:* $\lambda=\Theta(C_{\text{poly}}K^{1/(1+\beta)}+\log(K|\mathcal{P}|/\delta))$*;*
- *β-exponential decay:* $\lambda=\Theta(C_{\exp}(\log K)^{1/\beta}+\log(K|\mathcal{P}|/\delta))$*;*

c is an absolute constant, then with probability at least $1-\delta$, $\forall k\in[K]$ we have

$$V^{\pi^*,\widehat{\mathcal{P}}_k,r+\widehat{b}_k}-V^{\pi^*,\mathcal{P},r}\geqslant-\sqrt{|\mathcal{A}|^{L+1}\zeta_k}$$

Proof. With Lemma 14, we have that

$$
\begin{aligned}
&V^{\pi^*,\widehat{\mathcal{P}}_k,r+\widehat{b}^k}-V^{\pi^*,\mathcal{P},r} \\
=&\sum_{h\in[H]}\mathbb{E}_{(x_h,a_h)\sim d^{\pi^*}_{\widehat{\mathcal{P}}_k,h}}\left[\widehat{b}^k_h(x_h,a_h)+\mathbb{E}_{o'\sim\mathbb{P}^{\widehat{\mathcal{P}}_k}_h(\cdot|x_h,a_h)}[V^{\pi^*,\mathcal{P},r}_{h+1}(x'_{h+1})]-\mathbb{E}_{o'\sim\mathbb{P}^{\mathcal{P}}_h(\cdot|x_h,a_h)}[V^{\pi^*,\mathcal{P},r}_{h+1}(x'_{h+1})]\right] \\
\geqslant&\sum_{h\in[H]}\mathbb{E}_{(x_h,a_h)\sim d^{\pi^*}_{\widehat{\mathcal{P}}_k,h}}\left[\min\left[c\alpha_k\|\hat{p}(\cdot|x_{h-L},a_{h-L})\|_{L_2(\mu),\Sigma^{-1}_{\rho_{k,h-L},\hat{p}}},2\right]+\mathbb{E}_{o'\sim\mathbb{P}^{\widehat{\mathcal{P}}_k}_h(\cdot|x_h,a_h)}[V^{\pi^*,\mathcal{P},r}_{h+1}(x'_{h+1})]\right. \\
&\left.-\mathbb{E}_{o'\sim\mathbb{P}^{\mathcal{P}}_h(\cdot|x_h,a_h)}[V^{\pi^*,\mathcal{P},r}_{h+1}(x'_{h+1})]\right],
\end{aligned}
$$

where in the last step we replace the empirical covariance with the population counterpart thanks to Lemma 17 in Ren et al. (2023b). Define

$$g_h(z_h,a_h)=\mathbb{E}_{o'\sim\mathbb{P}^{\mathcal{P}}_h(\cdot|x_h,a_h)}[V^{\pi^*,\mathcal{P},r}_{h+1}(x'_{h+1})]-\mathbb{E}_{o'\sim\mathbb{P}^{\widehat{\mathcal{P}}_k}_h(\cdot|x_h,a_h)}[V^{\pi^*,\mathcal{P},r}_{h+1}(x'_{h+1})],$$

With Hölder's inequality, we have that $\|g_h\|_\infty\leqslant 2$.

Furthermore, with Lemma 10, we have that

$$\begin{aligned}
&\sum_{h\in[H]} \mathbb{E}_{(x_h,a_h)\sim d^{\pi^*}_{\widehat{\mathcal{P}}_k,h}}[g_h(x_h,a_h)] \\
\leqslant &\sum_{h\in[H]} \mathbb{E}_{(x_{h-L},a_{h-L})\sim d^{\pi^*}_{\widehat{\mathcal{P}}_k,h}} \left[\|\hat{p}(\cdot|x_{h-L},a_{h-L})\|_{L_2(\mu),\Sigma^{-1}_{\rho_{k,h-L},\hat{p}}}\right] \\
&\cdot \sqrt{k|\mathcal{A}|^L \cdot \mathbb{E}_{(\tilde{x}_h,\tilde{a}_h)\sim \rho_{h-2L}\circ^{2L}\mathcal{U}(\mathcal{A})}\left[g(\tilde{x}_h,\tilde{a}_h)^2\right] + 4\lambda C + 4kL|\mathcal{A}|^{L-1}\zeta_k} \\
\leqslant &\sum_{h\in[H]} \mathbb{E}_{(x_{h-L},a_{h-L})\sim d^{\pi^*}_{\widehat{\mathcal{P}}_k,h}} \left[c\alpha_k \|\hat{p}(\cdot|x_{h-L},a_{h-L})\|_{L_2(\mu),\Sigma^{-1}_{\rho_{k,h-L},\hat{p}}}\right],
\end{aligned}$$

where we use Lemma 15 in the last step. Now we deal with the case with $h \in [L]$. Note that, $\forall h \in [L]$

$$\begin{aligned}
&\mathbb{E}_{(x_h,a_h)\sim\pi}[g_h(x_h,a_h)] \\
\leqslant &|\mathcal{A}|^h \mathbb{E}_{x_1\sim d_1, a_{1:h}\sim\circ^h\mathcal{U}(\mathcal{A})} \left\|\mathbb{P}_h^{\widehat{\mathcal{P}}_k}(\cdot|x_h,a_h) - \mathbb{P}_h^{\mathcal{P}}(\cdot|x_h,a_h)\right\|_1 \\
\leqslant &\sqrt{\mathbb{E}_{x_1\sim d_1, a_{1:h}\circ^L\mathcal{U}(\mathcal{A})} \left\|\mathbb{P}_h^{\widehat{\mathcal{P}}_k}(\cdot|x_h,a_h) - \mathbb{P}_h^{\mathcal{P}}(\cdot|x_h,a_h)\right\|_1^2} \\
\leqslant &\sqrt{|\mathcal{A}|^h\zeta_k},
\end{aligned}$$

where in the last step we use Lemma 15. We finish the proof by summing over $h \in [L]$. □

Lemma 12 (Regret)**.** *With probability at least* $1-\delta$*, we have that*

- *For* β*-finite spectrum, we have*
$$\sum_{k=1}^K V^{\pi^*,\mathcal{P},r} - V^{\pi_k,\mathcal{P},r} \lesssim \sum_{k=1}^K V^{\pi^*,\mathcal{P},r} - V^{\pi_k,\mathcal{P},r} \lesssim H^2\beta^{3/2}|\mathcal{A}|^L \log K\sqrt{CLK\log(K|\mathcal{M}|/\delta)};$$
- *For* β*-polynomial decay, we have*
$$\sum_{k=1}^K V^{\pi^*,\mathcal{P},r} - V^{\pi_k,\mathcal{P},r} \lesssim C_{\text{poly}}H^2|\mathcal{A}|^L K^{\frac{1}{2}+\frac{1}{1+\beta}}\sqrt{CL\log(K|\mathcal{M}|/\delta)};$$
- *For* β*-exponential decay, we have*
$$\sum_{k=1}^K V^{\pi^*,\mathcal{P},r} - V^{\pi_k,\mathcal{P},r} \lesssim C_{\exp}H^2|\mathcal{A}|^L(\log K)^{1+\frac{3}{2\beta}}\sqrt{CLK\log(K|\mathcal{M}|/\delta)};$$

Proof. With Lemma 11 and Lemma 14, we have

$$\begin{aligned}
&V^{\pi^*,\mathcal{P},r} - V^{\pi_k,\mathcal{P},r} \\
\leqslant &V^{\pi^*,\widehat{\mathcal{P}}_k,r+\widehat{b}^k} + \sqrt{|\mathcal{A}|^{L+1}\zeta_k} - V^{\pi_k,\mathcal{P},r} \\
\leqslant &V^{\pi^k,\widehat{\mathcal{P}}_k,r+\widehat{b}^k} + \sqrt{|\mathcal{A}|^{L+1}\zeta_k} - V^{\pi_k,\mathcal{P},r} \\
= &\sum_{h\in[H]} \mathbb{E}_{(x_h,a_h)\sim d^{\pi_k}_{\mathcal{P},h}}\left[\widehat{b}^k_h(x_h,a_h) + \mathbb{E}_{o'\sim\mathbb{P}_h^{\widehat{\mathcal{P}}_k}(\cdot|x_h,a_h)}\left[V^{\pi_k,\widehat{\mathcal{P}}_k,r+\widehat{b}^k_h}_{h+1}(x'_{h+1})\right] - \mathbb{E}_{o'\sim\mathbb{P}_h^{\mathcal{P}}(\cdot|x_h,a_h)}\left[V^{\pi_k,\widehat{\mathcal{P}}_k,r+\widehat{b}^k_h}_{h+1}(x'_{h+1})\right]\right], \\
&+ \sqrt{|\mathcal{A}|^{L+1}\zeta_k}.
\end{aligned}$$

Note that $\left\|\widehat{b}^k_h\right\|_\infty \leqslant 2$. Applying Lemma 9, we have that

$$\begin{aligned}
&\sum_{h\in[H]} \mathbb{E}_{(x_h,a_h)\sim d^{\pi_k}_{\mathcal{P},h}}\left[\widehat{b}^k_h(x_h,a_h)\right] \\
\leqslant &\sum_{h\in[H]} \mathbb{E}_{(\tilde{x}_{h-L},\tilde{a}_{h-L})\sim d^{\pi_k}_{\mathcal{P},h}} \left[\|p^*(\cdot|x_{h-L},a_{h-L})\|_{L_2(\mu),\Sigma^{-1}_{\rho_{k,h-L},p^*}}\right] \\
&\cdot \sqrt{k|\mathcal{A}|^L \cdot \mathbb{E}_{(\tilde{x}_h,\tilde{a}_h)\sim\rho_{k,h-L}\circ^L\mathcal{U}(\mathcal{A})}\left[\widehat{b}^k_h(\tilde{x}_h,\tilde{a}_h)^2\right] + 4\lambda C}
\end{aligned}$$

Following the proof of Lemma 8 in Ren et al. (2023b), we have that:

- for β-finite spectrum,
$$k\mathbb{E}_{(\tilde{x}_h,\tilde{a}_h)\sim\rho_{k,h-L}\circ^L\mathcal{U}(\mathcal{A})}\left[\widehat{b}_h^k(\tilde{x}_h,\tilde{a}_h)^2\right]=O(\beta\log K);$$
- for β-polynomial decay,
$$k\mathbb{E}_{(\tilde{x}_h,\tilde{a}_h)\sim\rho_{k,h-L}\circ^L\mathcal{U}(\mathcal{A})}\left[\widehat{b}_h^k(\tilde{x}_h,\tilde{a}_h)^2\right]=O\left(C_{\text{poly}}K^{\frac{1}{2(1+\beta)}}\log K\right);$$
- for β-exponential decay,
$$k\mathbb{E}_{(\tilde{x}_h,\tilde{a}_h)\sim\rho_{k,h-L}\circ^L\mathcal{U}(\mathcal{A})}\left[\widehat{b}_h^k(\tilde{x}_h,\tilde{a}_h)^2\right]=O\left(C_{\exp}(\log K)^{1+1/\beta}\right).$$

We then consider
$$\sum_{h\in[H]}\mathbb{E}_{(x_h,a_h)\sim d^{\pi_k}_{\mathcal{P},h}}\left[\mathbb{E}_{o'\sim\mathbb{P}_h^{\widehat{\mathcal{P}}_k}(\cdot|x_h,a_h)}\left[V_{h+1}^{\pi_k,\widehat{\mathcal{P}}_k,r+\widehat{b}_h^k}(x'_{h+1})\right]-\mathbb{E}_{o'\sim\mathbb{P}_h^{\mathcal{P}}(\cdot|x_h,a_h)}\left[V_{h+1}^{\pi_k,\widehat{\mathcal{P}}_k,r+\widehat{b}_h^k}(x'_{h+1})\right]\right].$$
Define
$$g(x_h,a_h)=\frac{1}{2H+1}\left[\mathbb{E}_{o'\sim\mathbb{P}_h^{\widehat{\mathcal{P}}_k}(\cdot|x_h,a_h)}\left[V_{h+1}^{\pi_k,\widehat{\mathcal{P}}_k,r+\widehat{b}_h^k}(x'_{h+1})\right]-\mathbb{E}_{o'\sim\mathbb{P}_h^{\mathcal{P}}(\cdot|x_h,a_h)}\left[V_{h+1}^{\pi_k,\widehat{\mathcal{P}}_k,r+\widehat{b}_h^k}(x'_{h+1})\right]\right].$$
With Hölder's inequality and note that $\left\|\widehat{b}_h^k\right\|\leqslant 2$, we have that $\|g\|_\infty\leqslant 2$. With Lemma 9, we have that
$$\begin{aligned}
&\sum_{h\in[H]}\mathbb{E}_{(x_h,a_h)\sim d^{\pi_k}_{\mathcal{P},h}}[g(x_h,a_h)]\\
\leqslant&\sum_{h\in[H]}\mathbb{E}^{\mathcal{P}}_{(x_{h-L},a_{h-L})\sim d^{\pi_k}_{\mathcal{P},h}}\left[\|p^*(\cdot|x_{h-L},a_{h-L})\|_{L_2(\mu),\Sigma^{-1}_{\rho_{k,h-L},p^*}}\right]\cdot\sqrt{k|\mathcal{A}|^L\mathbb{E}_{(\tilde{x}_h,\tilde{a}_h)\sim\rho_{k,h-L}\circ^L\mathcal{U}(\mathcal{A})}\left[g(\tilde{x}_h,\tilde{a}_h)^2\right]+4\lambda C}\\
\leqslant&\sum_{h\in[H]}\mathbb{E}^{\mathcal{P}}_{(x_{h-L},a_{h-L})\sim d^{\pi_k}_{\mathcal{P},h}}\left[\|p^*(\cdot|x_{h-L},a_{h-L})\|_{L_2(\mu),\Sigma^{-1}_{\rho_{k,h-L},p^*}}\right]\cdot\sqrt{k|\mathcal{A}|^L\zeta_k+4\lambda C}\\
\leqslant&c\alpha_k\sum_{h\in[H]}\mathbb{E}^{\mathcal{P}}_{(x_{h-L},a_{h-L})\sim d^{\pi}_{\mathcal{P},h}}\left[\|p^*(\cdot|x_{h-L},a_{h-L})\|_{L_2(\mu),\Sigma^{-1}_{\rho_{k,h-L},p^*}}\right]
\end{aligned}$$
With Cauchy-Schwartz inequality, we know that
$$\begin{aligned}
&\sum_{k\in[K]}\mathbb{E}^{\mathcal{P}}_{(x_{h-L},a_{h-L})\sim d^{\pi_k}_{\mathcal{P},h}}\left[\|p^*(\cdot|x_{h-L},a_{h-L})\|_{L_2(\mu),\Sigma^{-1}_{\rho_{k,h-L},p^*}}\right]\\
\leqslant&\sqrt{K\sum_{k\in[K]}\mathbb{E}^{\mathcal{P}}_{(x_{h-L},a_{h-L})\sim d^{\pi_k}_{\mathcal{P},h}}\left[\|p^*(\cdot|x_{h-L},a_{h-L})\|^2_{L_2(\mu),\Sigma^{-1}_{\rho_{k,h-L},p^*}}\right]}.
\end{aligned}$$
Following the proof of Lemma 8 in Ren et al. (2023b), we have that

- for β-finite spectrum,
$$\sum_{k\in[K]}\mathbb{E}^{\mathcal{P}}_{(x_{h-L},a_{h-L})\sim d^{\pi_k}_{\mathcal{P},h}}\left[\|p^*(\cdot|x_{h-L},a_{h-L})\|^2_{L_2(\mu),\Sigma^{-1}_{\rho_{k,h-L},p^*}}\right]=O(\beta\log K);$$
- for β-polynomial decay,
$$\sum_{k\in[K]}\mathbb{E}^{\mathcal{P}}_{(x_{h-L},a_{h-L})\sim d^{\pi_k}_{\mathcal{P},h}}\left[\|p^*(\cdot|x_{h-L},a_{h-L})\|^2_{L_2(\mu),\Sigma^{-1}_{\rho_{k,h-L},p^*}}\right]=O\left(C_{\text{poly}}K^{\frac{1}{2(1+\beta)}}\log K\right);$$
- for β-exponential decay,
$$\sum_{k\in[K]}\mathbb{E}^{\mathcal{P}}_{(x_{h-L},a_{h-L})\sim d^{\pi_k}_{\mathcal{P},h}}\left[\|p^*(\cdot|x_{h-L},a_{h-L})\|^2_{L_2(\mu),\Sigma^{-1}_{\rho_{k,h-L},p^*}}\right]=O\left(C_{\exp}(\log K)^{1+1/\beta}\right).$$

Combine the previous steps and take the dominating term out, we have that

- for β-finite spectrum,
$$\sum_{k=1}^K V^{\pi^*,\mathcal{P},r}-V^{\pi_k,\mathcal{P},r}\lesssim H^2\beta^{3/2}|\mathcal{A}|^L\log K\sqrt{CLK\log(K|\mathcal{M}|/\delta)};$$

- for β-polynomial decay,
$$\sum_{k=1}^{K} V^{\pi^*,\mathcal{P},r} - V^{\pi_k,\mathcal{P},r} \lesssim C_{\text{poly}} H^2 |\mathcal{A}|^L K^{\frac{1}{2}+\frac{1}{1+\beta}} \sqrt{CL\log(K|\mathcal{M}|/\delta)};$$
- for β-exponential decay,
$$\sum_{k=1}^{K} V^{\pi^*,\mathcal{P},r} - V^{\pi_k,\mathcal{P},r} \lesssim C_{\text{exp}} H^2 |\mathcal{A}|^L (\log K)^{1+\frac{3}{2\beta}} \sqrt{CLK\log(K|\mathcal{M}|/\delta)};$$

which finishes the proof. □

Theorem 13 (PAC Guarantee)**.** *After interacting with the environments for KH episodes*

- $K = \Theta\left(\frac{CH^4L\beta^3|\mathcal{A}|^{2L}\log(|\mathcal{P}|/\delta)}{\varepsilon^2}\log^3\left(\frac{CH^4L\beta^3|\mathcal{A}|^{2L}\log(|\mathcal{P}|/\delta)}{\varepsilon^2}\right)\right)$ *for β-finite spectrum;*
- $K = \Theta\left(C_{\text{poly}}\left(\frac{H^2L|\mathcal{A}|^L\sqrt{C\log(|\mathcal{P}|/\delta)}}{\varepsilon}\log^{3/2}\left(\frac{\sqrt{C}H^2L|\mathcal{A}|^L\log(|\mathcal{P}|/\delta)}{\varepsilon}\right)\right)^{\frac{2(1+\beta)}{\beta-1}}\right)$ *for β-polynomial decay;*
- $K = \Theta\left(\frac{C_{\text{exp}}CH^4L|\mathcal{A}|^{2L}\log(|\mathcal{P}|/\delta)}{\varepsilon^2}\log^{\frac{3+2\beta}{\beta}}\left(\frac{CH^4L|\mathcal{A}|^{2L}\log(|\mathcal{P}|/\delta)}{\varepsilon^2}\right)\right)$ *for β-exponential decay;*

we can obtain an ε-optimal policy with high probability.

Proof. This is a direct extension of the proof of Theorem 9 in Ren et al. (2023b). □

G. Technical Lemma

Lemma 14 (Simulation Lemma)**.** *For two MDPs $\mathcal{M} = (P, r)$ and $\mathcal{M}' = (P', r + b)$, we have*
$$\begin{aligned} &V^{\pi}_{P',r+b} - V^{\pi}_{P,r} \\ &= \sum_{h\in[H]} \mathbb{E}_{(s_h,a_h)\sim d^h_{P,\pi}}\left[b_h(s_h,a_h) + \mathbb{E}_{s_{h+1}\sim P'(s_h,a_h)}\left[V^{\pi}_{P',r+b,h+1}(s_{h+1})\right] - \mathbb{E}_{s_{h+1}\sim P(s_h,a_h)}\left[V^{\pi}_{P',r+b,h+1}(s_{h+1})\right]\right], \end{aligned}$$
and
$$\begin{aligned} &V^{\pi}_{P',r+b} - V^{\pi}_{P,r} \\ &= \sum_{h\in[H]} \mathbb{E}_{(s_h,a_h)\sim d^h_{P',\pi}}\left[b_h(s_h,a_h) + \mathbb{E}_{s_{h+1}\sim P'(s_h,a_h)}\left[V^{\pi}_{P,r,h+1}(s_{h+1})\right] - \mathbb{E}_{s_{h+1}\sim P(s_h,a_h)}\left[V^{\pi}_{P,r,h+1}(s_{h+1})\right]\right], \end{aligned}$$

For the proof, see Uehara et al. (2021) for an example.

Lemma 15 (MLE Guarantee)**.** *For any episoode $k \in [K]$, step $h \in [H]$, define ρ_h as the joint distribution of (x_h, a_h) in the dataset $\mathcal{D}_{h,k}$ at episode k. Then with probability at least $1 - \delta$, we have that*
$$\mathbb{E}_{(x_h,a_h)\sim\mathcal{D}_{h,k}}\left\|\mathbb{P}^{\mathcal{P}}_h(\cdot|x_h,a_h) - \mathbb{P}^{\widehat{\mathcal{P}}_k}_h(\cdot|x_h,a_h)\right\|_1^2 \leqslant \zeta_k,$$
where $\zeta_k = O(\log(Hk|\mathcal{M}|/\delta)/k)$

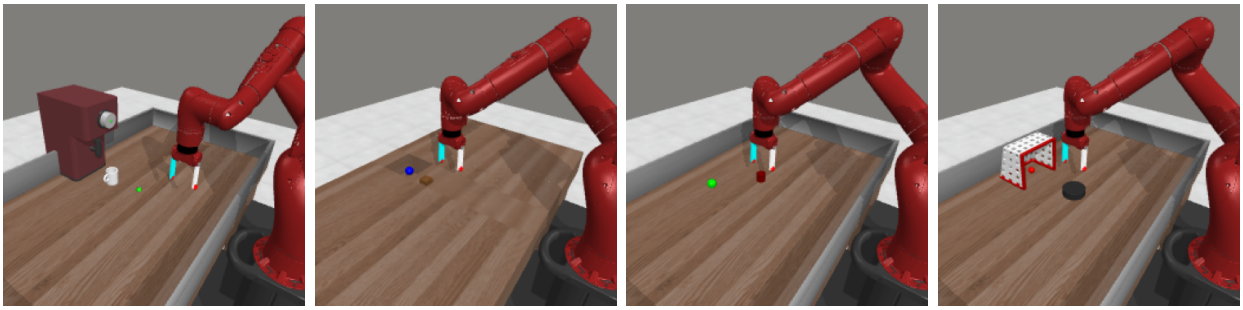
For the proof, see Agarwal et al. (2020).

H. Implementation Details on Image-based Continuous Control

We evaluate our method on Meta-world (Yu et al., 2019) [1] and DeepMind Control Suites (Tassa et al., 2018) [2] to demonstrate its capability for complex visual control tasks. Meta-world is an open-source simulated benchmark consisting of 50 distinct robotic manipulation tasks. The DeepMind Control Suite is a set of continuous control tasks with a standardized structure and interpretable rewards, intended to serve as performance benchmarks for reinforcement learning agents. The visualization

[1] https://github.com/Farama-Foundation/Metaworld
[2] https://github.com/google-deepmind/dm_control

of some tasks from the two domains are shown in Figures 3 and 4. With only one frame of the visual observation, we will miss some information related to the task, for example the speed, thus these tasks are partially observable. The performance for Meta-world tasks are shown in Figure 5.



(a) Coffee Pull (b) Hand Insert (c) Push (d) Plate Slide

Figure 3. Visualization of the visual robotic manipulation tasks in Meta-world.

(a) Acrobot Swingup (b) Reacher hard (c) Reach Duplo (d) Quadruped Run

Figure 4. Visualization of the visual control tasks in DeepMind Control Suites.

In particular, we employ visual observations with dimensions of $64 \times 64 \times 3$ and apply a variational autoencoder (VAE) to learn representations for these visual observations. The VAE is first pre-trained with random trajectories at the beginning and then fine-tuned during the online learning procedure. It produces compact vector representations for the images, which are then forwarded as input to our representation learning method. We apply actor-critic Learning based on the representation learned by VAE. The configuration of used tasks are given in Table 1. The hyperparameters used in the RL agent are shown in Table 2.

H.1. Ablation Studies

The importance of the exploration has been demonstrated in (Zhang et al., 2022). We perform ablation studies to demonstrate the effects of the major components, including representation dimension and window size, as illustrated below. Figure 6 presents an ablation study on representation dimension, where we compare μLV-Rep with latent representation dimensions 2048, 512, and 128. We also ablate the effect of window size L. In Figure 7, we compare μLV-Rep with window size $L = 1, 3, 5$. We also compare DrQ-v2 (Yarats et al., 2022) with $L = 1, 3, 5$ to show the effect of L on other algorithms. The results show that for $L = 1$, both μLV-Rep and DrQ-v2 struggle with learning, which confirms the Non-Markovian property of the DMC control problems. We can also find that $L = 3$ is sufficient for learning in both test domains.

H.2. Experiment of Partially Observable Continuous Control

We also evaluate the proposed approach on RL tasks with partial observations, constructed based on the OpenAI gym MuJoCo (Todorov et al., 2012). Standard MuJoCo tasks from the OpenAI gym and DeepMind Control Suites are not

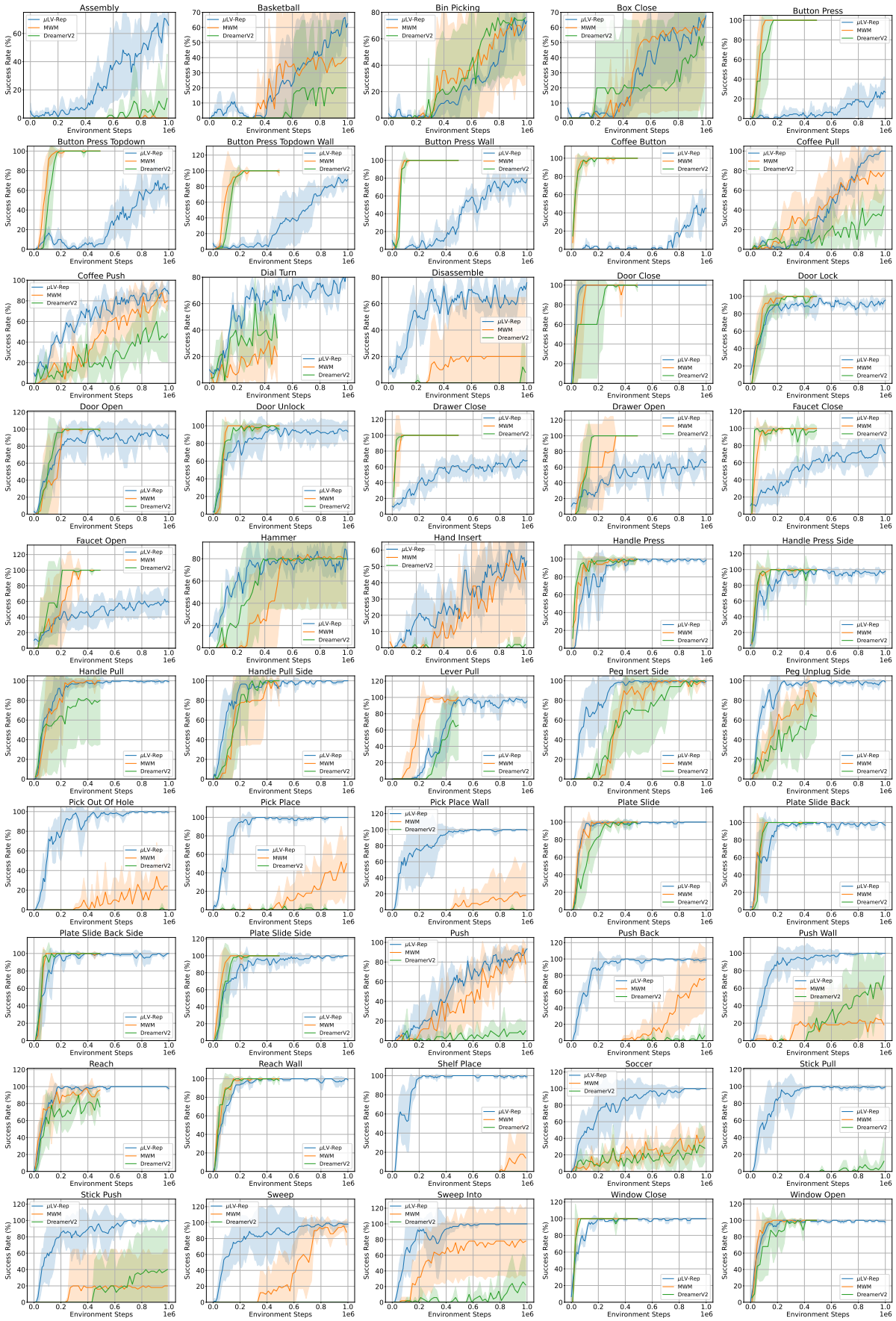


Figure 5. Overall performance on MetaWorld tasks.

Table 1. Configuration of environments.

Hyperparameter	Value
Image observation	$64 \times 64 \times 3$
Image normalization	Mean: $(0.485, 0.456, 0.406)$, Std: $(0.229, 0.224, 0.225)$
Action repeat	2
Episode length	500 (Meta-world), 1000 (DMC)
Normalize action	[-1,1]
Camera	corner2 (Meta-world), camera2 (DMC)
Total steps in environment	1M (Meta-world), 0.5M (DMC)

Table 2. Hyperparameters in μLV-Rep. The numbers in Conv and MLP denote the output channels and units.

Hyperparameter	Value
Buffer size	1,000,000
Batch size	256
Random steps	4000
Pretrain step	10000
Features dim.	100
Hidden dim.	1024
Encoder	(Conv(32), Conv(32), Conv(32), Conv(32), MLP(100))
Image Decoder	(MLP(100), MLP(1024), ConvT(32), ConvT(32), ConvT(32), ConvT(32), Conv(3))
Actor Network	(MLP(1024),MLP(1024),MLP(Action Space))
Critic Network	(MLP(1024),MLP(1024),MLP(1))
Optimizer	Adam
Learning rate	0.0001
Discount	0.99
Critic soft-update rate	0.01
Evaluate interval	10,000
Evaluate episodes	10 (Meta-world), 5 (DMC)

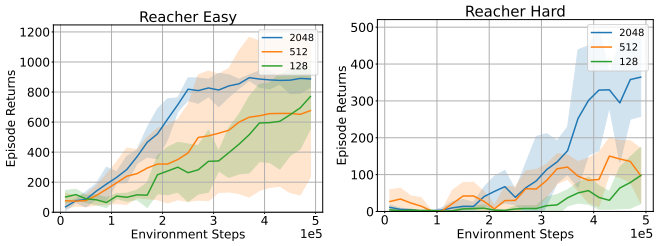


Figure 6. Ablation of feature dimension on visual control tasks from DeepMind Control Suites. Increasing the dimension of the feature gets better performance.

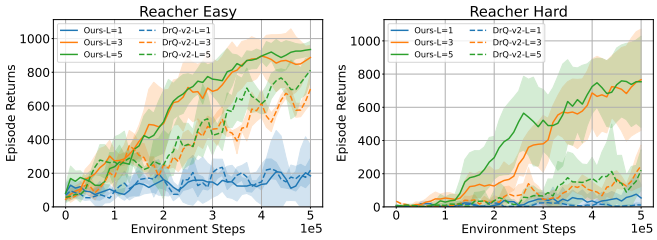


Figure 7. Ablation of window size L on visual control tasks from DeepMind Control Suites. $L = 3$ is sufficient for learning in both test domains.

Table 3. Performance on various continuous control problems with partial observation. All results are averaged across 4 random seeds and a window size of 10K. μLV-Rep achieves the best performance compared to the baselines. Here, Best-FO denotes the performance of LV-Rep using full observations as inputs, providing a reference on how well an algorithm can achieve most in our tests.

	HalfCheetah	Humanoid	Walker	Ant	Hopper
μLV-Rep	**3596.2** $\pm$ **874.5**	**806.7** $\pm$ **120.7**	**1298.1**$\pm$ **276.3**	**1621.4** $\pm$ **472.3**	**1096.4** $\pm$ **130.4**
Dreamer-v2	2863.8 $\pm$ 386	672.5 $\pm$ 36.6	**1305.8** $\pm$ **234.2**	1252.1 $\pm$ 284.2	758.3 $\pm$ 115.8
SAC-MLP	1612.0 $\pm$ 223	242.1 $\pm$ 43.6	736.5 $\pm$ 65.6	**1612.0** $\pm$ **223**	614.15 $\pm$ 67.6
SLAC	**3012.4** $\pm$ 724.6	387.4 $\pm$ 69.2	536.5 $\pm$ 123.2	1134.8 $\pm$ 326.2	739.3 $\pm$ 98.2
PSR	2679.75$\pm$386	534.4 $\pm$ 36.6	862.4 $\pm$ 355.3	1128.3 $\pm$ 166.6	818.8 $\pm$ 87.2
Best-FO	5557.6$\pm$439.5	1086$\pm$278.2	2523.5$\pm$333.9	2511.8$\pm$460.0	2204.8$\pm$496.0
	Cheetah-run	Walker-run	Hopper-run	Humanoid-run	Pendulum
μLV-Rep	525.3 $\pm$ 89.2	**702.3** $\pm$ **124.3**	**69.3**$\pm$ **12.8**	**9.8** $\pm$ **6.4**	**168.2** $\pm$ **5.3**
Dreamer-v2	**602.3** $\pm$ **48.5**	438.2 $\pm$ 78.2	**59.2** $\pm$ **15.9**	2.3 $\pm$ 0.4	**172.3** $\pm$ **8.0**
SAC-MLP	483.3 $\pm$ 77.2	279.8 $\pm$ 190.6	19.2 $\pm$ 2.3	1.2 $\pm$ 0.1	163.6 $\pm$ 9.3
SLAC	105.1 $\pm$ 30.1	139.2 $\pm$ 3.4	36.1 $\pm$ 15.3	0.9 $\pm$ 0.1	**167.3** $\pm$ **11.2**
PSR	173.7 $\pm$ 25.7	57.4 $\pm$ 7.4	23.2 $\pm$ 9.5	0.8 $\pm$ 0.1	159.4 $\pm$ 9.2
Best-FO	639.3$\pm$24.5	724.2$\pm$37.8	72.9$\pm$40.6	11.8$\pm$6.8	167.1$\pm$3.1

partially observable. To generate partially observable problems based on these tasks, we adopt a widely employed approach of masking velocities within the observations (Ni et al., 2021; Weigand et al., 2021; Gangwani et al., 2020). In this way, it becomes impossible to extract complete decision-making information from a single environment observation, yet the ability to reconstruct the missing observation remains achievable by aggregating past observations. We provide the best performance when using the original fully observable states (without velocity masking) as input, denoted by *Best-FO* (Best result with Full Observations). This gives a reference for the best result an algorithm is expected to achieve in our tests.

We consider four baselines in the experiments, including two model-based methods Dreamer (Hafner et al., 2020; 2021) and Stochastic Latent Actor-Critic (SLAC) (Lee et al., 2020), and a model-free baseline, SAC-MLP, that concatenates history sequences (past four observations) as input to an MLP layer for both the critic and policy. This simple baseline can be viewed as an analogue to how DQN processes observations in Atari games (Mnih et al., 2013) as a sanity check. We also compare to the neural PSR (Guo et al., 2018). We compare all algorithms after running 200K environment steps. This setup exactly follows the benchmark (Wang et al., 2019), which has been widely adopted in (Zhang et al., 2022; Ren et al., 2023c;b) for fairness. All results are averaged across four random seeds. Table 3 presents all the experimental results, averaged over four random seeds. The results clearly demonstrate that the proposed method consistently delivers either competitive or superior outcomes across all domains compared to both the model-based and model-free baselines. We note that in most domains, μLV-Rep nearly matches the performance of Best-FO, further confirming that the proposed method is able to extract useful representations for decision-making in partially observable environments.