Planning, Fast and Slow: Online Reinforcement Learning with Action-Free Offline Data via Multiscale Planners

Chengjie Wu [*1] **Hao Hu** [*1] **Yiqin Yang** [2] **Ning Zhang** [3] **Chongjie Zhang** [3]

Abstract

The surge in volumes of video data offers unprecedented opportunities for advancing reinforcement learning (RL). This growth has motivated the development of passive RL, seeking to convert passive observations into actionable insights. This paper explores the prerequisites and mechanisms through which passive data can be utilized to improve online RL. We show that, in identifiable dynamics, where action impact can be distinguished from stochasticity, learning on passive data is statistically beneficial. Building upon the theoretical insights, we propose a novel algorithm named Multiscale State-Centric Planners (MSCP) that leverages two planners at distinct scales to offer guidance across varying levels of abstraction. The algorithm's fast planner targets immediate objectives, while the slow planner focuses on achieving longer-term goals. Notably, the fast planner incorporates pessimistic regularization to address the distributional shift between offline and online data. MSCP effectively handles the practical challenges involving imperfect pretraining and limited dataset coverage. Our empirical evaluations across multiple benchmarks demonstrate that MSCP significantly outperforms existing approaches, underscoring its proficiency in addressing complex, long-horizon tasks through the strategic use of passive data.

1. Introduction

Reinforcement Learning (RL) (Sutton & Barto, 2018) has exhibited remarkable scalability with large amounts of data in effectively addressing challenging tasks in video games (Reed et al., 2022; Bauer et al., 2023; Baker et al., 2022; Kumar et al., 2023), robotics manipulation (Seo et al., 2022; Chebotar et al., 2023), and scientific discovery (Jumper et al., 2021; Fawzi et al., 2022). Offline RL (Levine et al., 2020) holds great promise as it enables policy learning from pre-collected datasets, thus circumventing the expensive and sometimes infeasible exploration typically associated with online RL (Fujimoto et al., 2019; Prudencio et al., 2022). However, offline RL still requires an annotated dataset with action and reward labels, and fails to exploit other unannotated datasets that are abundant and cheaper in the real world, such as videos (Zhu et al., 2023; Wu et al., 2023b).

Inspired by recent success of unsupervised pretraining in natural language processing (Brown et al., 2020) and computer vision (He et al., 2022), passive RL (Seo et al., 2022; Zhu et al., 2023) aims to leverage an action-and-reward-free dataset for pretraining to faciliate downstream online RL. Passive RL helps extract the valuable information in passive video-like observations and transform them into actionable insights, receiving great attention from the community.

To understand when and how passive data is useful, we conduct a rigorous analysis on the statistical benefit of passive data in linear MDPs. We show that when the effect of action is distinguishable from intrinsic stochasticity, passive data can reduce the effective horizon of the problem by providing dense training signals and improve learning efficiency accordingly. However, previous methods still fall short in solving challenging long-horizon tasks in online learning, even equipped with passive data.

We identify two significant challenges of passive RL. First, the extrapolation error during the online stage can be severe due to the limited dataset coverage. Specifically, the data collected during online exploration may significantly deviate from the passive dataset. Compared with standard offline-to-online settings (Nakamoto et al., 2023), this problem is exacerbated in passive RL because no policy can be learned offline to initialize and regularize the behavior due to lack of action labels. The second challenge involves the extraction and utilization of information from the passive dataset at various levels of abstraction. This is crucial for guiding the online policy to learn in a manner that balances

[*]Equal contribution [1]Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China [2]Institute of Automation, Chinese Academy of Sciences, China [3]Department of Computer Science & Engineering, Washington University in St. Louis, MO, USA. Correspondence to: Chengjie Wu <wucj19@mails.tsinghua.edu.cn>.

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

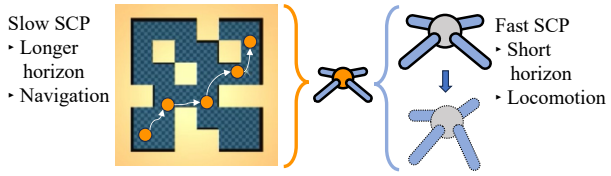

Figure 1. An *AntMaze* example to explain MSCP. The fast SCP plans immediate next states and provides guidance on locomotion, while the slow SCP proposes appropriate subgoals for navigation.

immediate and long-term objectives. A policy overly focused on immediate goals may result in a myopic policy, failing to achieve objectives that require a longer horizon. Conversely, an exclusive emphasis on the long-term objective may excessively complicate the exploration process.

In response to these challenges, we propose a novel algorithm termed **Multiscale State-Centric Planners (MSCP)**. We consider the goal-conditioned setting to remove the reliance on reward labels (Park et al., 2023). MSCP entails the pretraining of two state-centric planners (SCPs), each functioning at a distinct scale, emphasizing short-term and long-term objectives, respectively. As shown in Figure 1, the fast SCP proposes the immediate next state while the slow SCP focuses on planning subgoals at an appropriate distance. In online RL, the fast SCP forces pessimistic regularization by encouraging the policy to explore within the range of the dataset, mitigating the extrapolation error arising from the distributional shift from offline to online. The multiscale SCPs, planning at distinct levels, provide comprehensive guidance for online policy learning.

Empirically, we evaluate the effectiveness of our algorithm MSCP in four challenging environments that prioritize long-term planning: the *AntMaze* and *Kitchen* environment in D4RL (Fu et al., 2020), *CALVIN* (Mees et al., 2022), and *Procgen Maze* (Cobbe et al., 2020). The results show that MSCP significantly outperforms baselines, showcasing its ability to extract valuable information from passive dataset. The code for this paper is available at https://github.com/ChengjieWU/MSCP.

1.1. Related Work

Offline RL and Offline-to-Online RL Offline RL learns from a dataset with action labels, mainly addressing the extrapolation error in value estimation. Some methods enforce the trained policy to be close to the behavior policy via KL-divergence (Peng et al., 2019; Nair et al., 2020; Siegel et al., 2020; Wu et al., 2019; Yang et al., 2021; Ma et al., 2021). Other methods attempt to enforce a regularization constraint to penalize over-generalization (Kumar et al., 2020; Agarwal et al., 2020; Kostrikov et al., 2021; Hu et al., 2022). Offline-to-Online RL, on the other hand, aims to improve the policy trained offline by incorporating online RL. Different methods have been proposed, such as extracting high-level skills (Gupta et al., 2020; Ajay et al., 2020; Yang et al., 2023b), ensuring a smooth offline-to-online transition through expansion scheme (Zhang et al., 2023), ensembles(Lee et al., 2022; Ball et al., 2023), and calibrating value function (Nakamoto et al., 2023). Wu et al. (2023a) study the online adaptation of an offline learned exploitation policy in multi-agent games. There are also works exploring the semi-supervised setting to reuse the dynamical (Hu et al., 2023; Yu et al., 2022) and behavioral information (Hu et al., 2024; Park et al., 2024) in the reward-free data. Passive RL is more challenging as it cannot learn any policy offline due to the lack of both reward and action labels.

Passive RL. Passive RL consists of pretraining on passive dataset and subsequent online RL, also termed as pretraining from videos (Seo et al., 2022; Wu et al., 2023b; Ye et al., 2023). Zhu et al. study a similar setting that assumes access to rewards (2023). However, since the absence of rewards can be readily circumvented by adopting a goal-conditioned framework, we refrain from distinguishing between passive RL and action-free RL. We specifically address the challenges posed by the action-free scenario throughout the paper. The essence of passive RL lies in what is learned from the dataset and how it is learned. Some model-based methods focus on pretraining for cross-domain tasks, and partially pretrain a stacked network to leverage passive information (Seo et al., 2022; Wu et al., 2023b). Others address the single-domain scenario, where the same MDP is used in pretraining and online RL. FICC (Ye et al., 2023) maps the real action to the most common pretrained latent action, but cannot handle continuous action spaces in our evaluated environments. ICVF (Ghosh et al., 2023) pretrains state and goal representations to expedite downstream RL. Similar to our work, AF-Guide (Zhu et al., 2023) and HIQL (Park et al., 2023) also learn to plan in the state space. AF-Guide learns an action-free decision transformer to propose the immediate next state and also uses L2 distance as intrinsic reward. HIQL learns a high-level policy predicting subgoals at medium distance to provide a hierarchical decomposition when extracting policy from a pretrained value function. However, these works only use one level of planning and struggle with solving long-horizon tasks in passive RL.

To clarify, in the papers of ICVF and HIQL, their evaluation was conducted in the pure offline setting, wherein they trained exclusively on a large action-free dataset and a smaller dataset with actions, assumed to be of the same distribution[1]. When applied to passive RL, their performances significantly declines (see Table 1 and Figure 3), showcasing their susceptibility to extrapolation errors arising from the distributional shift from offline to online.

[1]Previous works generally sample a portion of the action-free dataset and reveal the action labels to form the annotated dataset

Hierarchical RL. Hierarchical RL and hierarchical imitation learning have shown promise in solving complex long-horizon problems by decomposing task into subtasks. HIRO (Nachum et al., 2018) and HAC (Levy et al., 2019) use a high level policy to predict future states, serving as objectives for the lower-level control policy. They use off-policy correction and hindsight relabelling to enhance training stability, respectively. Some other works learn different low-level skills represented by latent vectors, and learn a high-level controller to select appropriate skills (Bacon et al., 2017; Vezhnevets et al., 2017). Our method MSCP also employs a hierarchical structure using states as goals. However, we train two state planners predicting goals at different distances to extract both long-term and short-term planning abilities from an action-free dataset. These two planners collaborate to steer the training of the low-level control policy.

State-Based Imitation Learning. In state-based imitation learning, the action labels are also absent. However, it assumes expert demonstrations while passive RL does not. A state-based imitation learning method, SOIL (Radosavovic et al., 2021), trains an inverse dynamics model (IDM) and uses it to predict actions for the state-only demonstrations. A policy can then be trained with this action-annotated dataset. These two models are trained jointly. SAIL (Liu et al., 2020) reconstructs the expert policy through learning an IDM and a VAE that predicts the next state in the expert demo. To mitigate error accumulation, SAIL additionally uses RL to minimize the Wasserstein distance between states in the expert demo and the current trajectory. I2L (Gangwani & Peng, 2020) maintains a replay buffer storing online trajectories closest to the expert state-only demo, and updates the policy by mimicking these trajectories. Alternating between replay buffer update and policy learning, I2L gradually approaches the expert policy. In contrast, our method MSCP does not rely on expert demos and or learn an IDM. It learns a goal-conditioned V and state planners from a non-optimal action-free dataset. These models are then leveraged to efficiently train a low-level policy in the online stage.

2. Preliminary

We consider the Passive Reinforcement Learning (passive RL) setting, where online reinforcement learning is preceded by offline pretraining on passive data. We adopt the goal-conditioned RL (Schaul et al., 2015; Andrychowicz et al., 2017) formulation since it circumvents the need for reward labels. Passive RL is characterized by a goal-conditioned MDP $\mathcal{M}$ (Sutton & Barto, 2018) and an action-and-reward-free dataset $\mathcal{D}$. The MDP $\mathcal{M}$ is a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{G}, \mathcal{P}, r, \gamma)$, consisting of state space $\mathcal{S}$, action space $\mathcal{A}$, transition function $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, reward function $r : \mathcal{S} \times \mathcal{G} \to [0, r_{\max}]$, and discount factor $\gamma \in [0, 1)$.

We assume $\mathcal{S} = \mathcal{G}$ for the rest of the paper, as adopted by prior works (Park et al., 2023). The passive dataset $\mathcal{D}$ comprises solely of state sequences $\{s_0, s_1, \ldots, s_N\}$ collected by some unknown behavior policy in the MDP.

A policy $\pi : \mathcal{S} \times \mathcal{G} \to \Delta(\mathcal{A})$ specifies a decision-making strategy to select action $a \sim \pi(\cdot \mid s, g)$. The value function is defined as $V^\pi(s, g) = \mathbb{E}_\pi[\sum_{t=0}^\infty \gamma^t r(s_t, g) \mid s_0 = s]$ and the Q function is $Q^\pi(s, a, g) = \mathbb{E}_\pi[\sum_{t=0}^\infty \gamma^t r(s_t, g) \mid s_0 = s, a_0 = a]$. Additionally, we define a ***k*-distance state centric planner** (***k*-SCP**) to be a state space policy $\pi^{(k)}(\cdot|s, g) : \mathcal{S} \times \mathcal{G} \to \Delta(\mathcal{S})$ that predicts the state to be reached after k steps, conditioned on current state s and ultimate goal g. The Bellman operator is defined as:

$$(\mathbb{B}f)(s, a, g) = \mathbb{E}_{s' \sim p(\cdot|s,a)}\big[r(s, g) + \gamma f(s', g)\big]. \quad (1)$$

To characterize the value backup at the absence of actions, we also consider the following Bellman state-state operator:

$$(\mathbb{T}f)(s, s', g) = r(s, g) + \gamma f(s', g). \quad (2)$$

Suboptimality and online regret. We define the suboptimality of policy π as $\text{SubOpt}(\pi, s, g) = V^{\pi^\star}(s, g) - V^\pi(s, g)$, where $\pi^\star$ is the optimal policy under goal g. We consider the cumulative regret as the performance metric in theoretical analysis, which calculates the cumulative suboptimality for T timesteps:

$$\text{Reg}(T) = \sum_{t=1}^{T} \text{SubOpt}(\pi_t, s_t, g). \quad (3)$$

2.1. Linear MDP

To make things more concrete in theoretical analysis, we consider the *linear MDP* (Yang & Wang, 2019; Jin et al., 2020) as follows, where the transition kernel and expected reward function are linear with respect to a feature map.

Definition 2.1 (Linear MDP)**.** We say a MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{G}, \mathcal{P}, r, \gamma)$ is a linear MDP with known feature maps $\psi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ and $\varphi : \mathcal{S} \to \mathbb{R}^d$ if there exist unknown measures $\mu = (\mu_1, \ldots, \mu_d)$ over $\mathcal{S}$ and an unknown vector $\theta_g \in \mathbb{R}^d$ for each $g \in \mathcal{G}$ such that, $\mathcal{P}(s' \mid s, a) = \langle \psi(s, a), \mu(s') \rangle$ and $r(s, g) = \langle \varphi(s), \theta_g \rangle$ for all $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$. And we assume $\|\psi(s, a)\|_2 \le 1, \|\varphi(s, a)\|_2 \le 1$ for all $(s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ and $\max\{\|\mu(\mathcal{S})\|_2, \|\theta\|_2\} \le \sqrt{d}$, where $\|\mu(\mathcal{S})\| \equiv \int_{\mathcal{S}} \|\mu(s)\| ds$.

With the existence of a feature map, we can define the coverage coefficient in the feature space as follows.

Definition 2.2 (Goal-Conditioned Coverage Coefficient)**.** The goal-conditioned coverage coefficient $C^\dagger_{\mathcal{G}}$ with respect to the dataset $\mathcal{D}$ and the goal set $\mathcal{G}$ is defined as the supremum of C such that the following holds for all $s \in \mathcal{S}, g \in \mathcal{G}$ with probability at least $1 - \xi/2$:

$$\mathbb{E}_{\pi^\star_g}\big[\phi_t \phi_t^\top \,\big|\, s_0 = s\big] \succeq C \cdot \mathbb{E}_{\mathcal{D}}\big[\phi_t \phi_t^\top \,\big|\, s_0 = s\big], \quad (4)$$

where ξ is the confidence level, $\phi_t = \phi(s_t, s_{t+1})$ is the feature vector associated with the transition (s_t, s_{t+1}) defined in Lemma F.4, $\pi_g^\star$ is the optimal policy conditioned on g.

Definition 2.2 generalizes the coverage coefficient in offline RL (Jin et al., 2020; 2021). It signifies the highest proportion between the dataset distribution density and the density induced by the optimal policy. Intuitively, a larger $C_{\mathcal{G}}^\dagger$ reflects better dataset quality. It is equivalent to the standard coverage coefficient given a fixed g.

2.2. Action-Free Offline RL

IQL (Kostrikov et al., 2022) learns value function and policy from an offline dataset annotated with action labels. To address the passive dataset, a goal-conditioned action-free variant of IQL (GCAF-IQL) (Xu et al., 2022; Ghosh et al., 2023; Park et al., 2023) learns a goal-conditioned value function $V(s, g)$ (parameterized by θ_V) by minimizing:

$$\mathcal{L}_{\mathcal{D}}^V(\theta_V) = \mathbb{E}_{\mathcal{D}}\Big(L_2^\tau\big[r(s_t, g) + \gamma \bar{V}(s_{t+1}, g) - V(s, g)\big]\Big) \tag{5}$$

where $L_2^\tau(x) = |\tau - \mathbb{I}(x < 0)|x^2$ is expectile loss with parameter τ, and $\bar{V}$ is a target network periodically updated towards V at a slow rate. Similar to IQL, it uses expectile regression to approximate policy improvement within the the dataset's range. Additionally, treating states as actions, any k-SCP $\pi^{(k)}$ (parameterized by $\theta_{\pi^{(k)}}$) can also be extracted similarly with AWR-style objective (Peng et al., 2019):

$$\mathcal{J}_{\mathcal{D}}^{(k)}(\theta_{\pi^{(k)}}) = \mathbb{E}_{\mathcal{D}}\left[\exp\left(\beta \cdot \widetilde{A}^{(k)}(s_t, s_{t+k}, g)\right) \log \pi^{(k)}(s_{t+k}|s_t, g)\right], \tag{6}$$

where β is the inverse temperature hyperparameter, and $\widetilde{A}^{(k)}(s_t, s_{t+k}, g) = \sum_{i=t}^{t+k-1} r(s_i, g) + \gamma^k V(s_{t+k}, g) - V(s_t, g)$ is the advantage. Finally, if action labels become available, the trained $V(s, g)$ can also be used to extract the low-level control policy $\pi^l(\cdot|s, g)$ (parameterized by θ_{π^l}) by maximizing the objective:

$$\mathcal{J}_{\mathcal{D}}^l(\theta_{\pi^l}) = \mathbb{E}_{\mathcal{D}}\left[\exp\left(\beta \widetilde{A}^l(s_t, s_{t+1}, g)\right) \log \pi^l(a_t|s_t, g)\right] \tag{7}$$

where $\widetilde{A}^l(s_t, s_{t+1}, g) = r(s_t, g) + \gamma V(s_{t+1}, g) - V(s_t, g)$. HIQL (Park et al., 2023) further conditions the policy π^l on a closer subgoal predicted by a trained h-SCP, instead of the ultimate goal g. This hierarchical decomposition aims to reduce the noise in advantage estimation and improve performance. HIQL also shows that r and γ can be absorbed into hyperparameter β when the rewards are mostly constants, simplifying the calculation of the advantages: $\widetilde{A}^{(k)}(s_t, s_{t+k}, g) = V(s_{t+k}, g) - V(s_t, g)$, $\widetilde{A}^l(s_t, s_{t+1}, g) = V(s_{t+1}, g) - V(s_t, g)$.

3. Theoretical Analysis

What is the statistical benefit of passive data? Intuitively, passive data contains rich information of environment's dynamics. However, due to the lack of action labels, it may be impossible to disentangle the effects of actions from the stochasticity in the dynamics (Yang et al., 2023a; Park et al., 2023). Confounding the two factors can lead to potential value overestimation. This motivates us to study identifiable dynamics where the set of outcomes of actions are known.

Definition 3.1 (Identifiable Dynamics)**.** The dynamics $\mathcal{P}(s'|s, a)$ are identifiable if for all $s \in \mathcal{S}$, there exists a known outcome set $\Phi(s) = \{P_z(\cdot|s) \,|\, z \in \mathcal{Z}\}$, where $\mathcal{Z}$ is a abstract action space, such that for all $a \in \mathcal{A}$, there exists some z such that $P_z(s'|s) = \mathcal{P}(s'|s, a)$, and for all $z \in \mathcal{Z}$, there exists some action a such that $P_z(s'|s) = \mathcal{P}(s'|s, a)$.

Deterministic dynamics are always identifiable. The identifiability condition is broader than the deterministic assumption used in previous studies (Park et al., 2023; Ghosh et al., 2023). For a detailed explanation and examples of identifiability, please refer to Appendix D. Identifiable dynamics enable us to estimate latent dynamics using passive data. We find that, despite the lack of an explicit mapping from abstract action z to real action a, we can still perform Bellman updates to learn the value function without overestimation. Based on these insights, we design an algorithm as shown in Algorithm 1, with a more detailed version in Algorithm 4. The algorithm consists of an offline phase where a pessimistic goal-conditioned value function is learned from passive data, and an online phase where the policy efficiently learns from dense feedbacks provided by the pretrained value function.

Learning Passive Value Functions. We use passive data to learn a goal-conditioned value function $\widehat{V}(s, g)$ via value iteration in the abstract action space $\mathcal{Z}$. At each iteration, we approximate the Bellman state-state operator $\mathbb{T}$ with $\widehat{\mathbb{T}}$ by minimizing the TD loss, with details in Equation 15. Then we construct a penalty $\Gamma(s, s', g)$ to account for the uncertainty in the dynamics to remain pessimistic over the value function. With high probability, the difference between the true Bellman update $\mathbb{T}\widehat{V}(s, g)$ and the approximate Bellman update $\widehat{\mathbb{T}}\widehat{V}(s, g)$ is bounded by Γ. Then the pessimistic Q function in the abstract action space is computed as

$$\widehat{Q}(s, z, g) = \mathbb{E}_{s' \sim P_z(\cdot|s)}\left[\widehat{\mathbb{T}}\widehat{V}(s, g) - \Gamma(s, s', g)\right] \tag{9}$$

While we don't know the exact mapping between z and a, we can still compute the value function by taking the maximum over z as follows

$$\widehat{V}(s, g) = \max_{z \in \mathcal{Z}} \widehat{Q}(s, z, g). \tag{10}$$

Online learning. An optimal value function can provide

Algorithm 1 Online Learning with Passive Value Iteration

Input: Passive dataset $\mathcal{D} = \{s_1, s_2, \cdots, s_N\}$, potential goals $\mathcal{G}$, parameter ζ

▷ Offline Phase
for every g in $\mathcal{G}$ **do**
  Set $\widehat{V}(s, g) \leftarrow 0$ and construct the negative bonus $\Gamma(s, s', g)$ according to Equation 13.
  while not converged **do**
    Compute Bellman update as in Equation 9.
    Update value function $\widehat{V}(s, g)$ as in Equation 10.
  end while
end for
▷ Online Phase
Receive the target goal g and initial state s_1.
Randomly initialize π_1.
for $t = 1, \ldots, T$ **do**
  Execute $a_t \sim \pi_t(\cdot|s_t, g)$.
  Receive reward $r(s_t, g)$ and compute the approximate advantage:

$$\widehat{A}(s_t, a_t, g) = r(s_t, g) + \gamma \widehat{V}(s_{t+1}, g) - \widehat{V}(s_t, g). \tag{8}$$

  Compute the optimistic advantage function $\widetilde{A}_t$ according to Equation 19.
  Update policy π_{t+1} to maximize $\widetilde{A}_t(a, s, g)$.
end for

dense feedbacks for any transition. Therefore, we can apply bandit algorithms to learn the optimal policy, where we use the advantage function $\widehat{A}(s_t, a_t, g)$ with respect to the offline learned value function as the supervison. We construct an optimistic advantage function $\widetilde{A}_t$ with a bonus for exploration as described in Equation 19. The policy learns by maximizing $\widetilde{A}_t$.

3.1. Theoretical Guarantees

In this section, we provide the theoretical guarantees of Algorithm 1. Similar to standard offline reinforcement learning, the goal-conditioned value function learned from passive data has the following guarantee.

Theorem 3.2 (Estimation Error of Goal-Conditioned Value Function)**.** *Consider a linear MDP with identifiable dynamics. Suppose the passive dataset $\mathcal{D}$ have positive coverage coefficients $C^\dagger_{\mathcal{G}}$, then the offline learned value function $\widehat{V}(s, g)$ in Equation 10 satisfies, for all $s \in \mathcal{S}, g \in \mathcal{G}$,*

$$|\widehat{V}(s, g) - V^\star(s, g)| \leq \frac{2cr_{\max}}{(1-\gamma)^2}\sqrt{\frac{d^2\zeta}{NC^\dagger_{\mathcal{G}}}}$$

with probability $1 - \xi$, where ζ are logarithmic factors and c is an absolute constant.

See Appendix F.1 for detailed proof. Theorem 3.2 indicates that when the dynamics are identifiable, we can learn an approximate *state* value function $\widehat{V}(s, g)$ as in standard offline RL despite the lack of *action* labels. Consequently, the online regret of Algorithm 1 can be upper bounded by the following theorem.

Theorem 3.3. *The online regret of Algorithm 1 is upper bounded by*

$$\mathrm{Reg}(T) \leq \underbrace{\frac{2cr_{\max}}{(1-\gamma)^2}\sqrt{\frac{d^2\zeta_1}{NC^\dagger_{\mathcal{G}}}}T}_{\text{offline error}} + \underbrace{\frac{2\sqrt{d^2\zeta_2}\cdot r_{\max}}{1-\gamma}\sqrt{T}}_{\text{online error with reduced horizon}},$$

where ζ_1 and ζ_2 are logarithmic factors.

See Appendix F.2 for the proof. Theorem 3.3 provides a clear decomposition of the suboptimality exhibited by Algorithm 1 into two distinct terms: the online exploration cost and the offline estimation error. The estimation error stems from the bias in learning from a finite-sample dataset as discussed in Theorem 3.2, while the online term captures the cost associated with exploration. Compared to online learning without passive data, the online error term is reduced by a factor of $1 - \gamma$. In scenarios where the passive data is abundant ($N \gg T$), as frequently observed in real-world settings, the offline error term becomes negligible. Consequently, our algorithm exhibits significantly reduced regret in comparison to naive online learning.

Corollary 3.4. *When $N \geq C^\dagger_{\mathcal{G}} T/\gamma^2$, the regret bound of Algorithm 1 is smaller than pure online learning.*

See Appendix F.3 for the proof. To summarize, we provide an explanation of intuitions behind the theoretical analysis. We first show that we can estimate the goal-conditioned value function accurately with pure offline data (Theorem 3.2). This is because the state value function is not a function of actions and we can estimate it using standard offline RL. Then we can reuse this value function to reduce the problem horizon. By providing an immediate feedback using $\widehat{V}(s_{t+1}, g)$, the regret does not depend on the horizon of the problem and the regret bound is improved as shown in Theorem 3.3. In essence, sufficient amount of passive data can reduce the effective horizon of the problem, enhancing the efficiency of online learning.

4. Multiscale State Centric Planners (MSCP)

The theorem suggests that in identifiable dynamics, learning a goal-conditioned value function $V(s, g)$ from passive data is a general and provably beneficial way for subsequent online RL. The pretrained value function reduces the policy horizon and provides dense supervision signals. It is worth noting that the goal-conditioned action-free variant of IQL (GCAF-IQL), introduced in Section 2.2, can be regarded as an empirical counterpart of Algorithm 1. In contrast to the

optimistic Bellman operator, the use of expectile regression in GCAF-IQL approximates the pessimistic value iteration. The advantage weighted regression method approximates the one-step maximization of advantages in Algorithm 1.

However, when designing an efficient empirical algorithm, it is important to consider that the pretrained value function may not be optimal due to optimization difficulties and limited dataset coverage. Specifically, $V(s, g)$ may make erroneous generalization on out-of-distribution (OOD) states encountered during online exploration. Additionally, since the advantage in Equation 8 depends on both states s_t, s_{t+1} and goal g, conditioning on subgoals at varying distances allows guidance signals at different abstract levels be extracted from the pretrained V to facilitate policy learning.

Based on these insights, we propose a novel algorithm called Multiscale State Centric Planners (MSCP). Similar to Park et al. (2023), we consider the same state space $\mathcal{S}$ as the goal space $\mathcal{G}$, focusing on solving the goal-reaching problem. The reward function $r(s, g)(s, g \in \mathcal{S})$ is defined as 0 if $s = g$, and -1 otherwise. We first introduce the pretraining of MSCP, which includes learning a goal conditioned $V(s, g)$ and a fast SCP and a slow SCP, operating at distinct scales. Then we explain how the fast SCP serves as a pessimistic regularizer to mitigate the distributional shift caused by the transition from offline to online. Finally, we describe the overall online procedure of MSCP that efficiently leverages multiscale guidance.

4.1. Value Pretrain and SCP Extraction

As explained above, MSCP learns a goal-conditioned value function $V(s, g)$ (parameterized by θ_V) with GCAF-IQL by minimizing the loss function in Equation 5. Meanwhile, we extract a fast 1-SCP π^f that plans immediate next states, and a slow h-SCP π^s that plans at horizon h, parameterized by θ_f, θ_s repectively, by maximizing the objective in Equation 6, setting $k = 1$ and $k = h$ respectively. The SPCs will be used to better guide online policy learning.

4.2. Fast SCP as Pessimistic Regularizer

In online RL, if we train control policy $\pi^l(\cdot|s, g)$ (parameterized by θ_l) solely from the advantage weighted regression with respect to the advantage estimates $\widetilde{A}^l(s_t, s_{t+1}, g) = V(s_{t+1}, g) - V(s_t, g)$, as previous works such as POR (Xu et al., 2022) and HIQL (Park et al., 2023), the extrapolation error in value estimates will sevely hamper the learning of π^l. For instance, if π^l explores a state s that is not present in the dataset, the pretrained V may predict an arbitrary value, rendering the advantage estimate ineffective.

We propose levaraging a pretrained 1-SCP π^f to provide immediate pessimistic guidance for π^l. An AWR-style objective (Equation 6) constrains the SCP to plan within the support of the dataset. Given that the fast SCP π^f is well trained, its predictions will always stay within the dataset as long as the input s_t is in the dataset. Therefore, we can use π^f to plan the next state and construct an intrinsic reward to encourage π^l to explore with conservatism. Let $\widetilde{s}_{t+1}$ denote the predicted next state of π^f. We define the intrinsic reward as $r^f(s_{t+1}, \widetilde{s}_{t+1}) = -\|s_{t+1} - \widetilde{s}_{t+1}\|_2$. To avoid interfering with the pretrained V, which accounts for goal-reaching returns, we train a separate value function $V^f(s_t, g)$ online to fit the expected one-step intrinsic reward: $\mathbb{E}_{(s_t, s_{t+1})\sim\pi} r^f(s_{t+1}, \widetilde{s}_{t+1})$. For a mini-batch $\mathcal{B}$, the loss function is:

$$\mathcal{L}_{\mathcal{B}}^{V^f}(\theta_{V^f}) = \mathbb{E}_{\mathcal{B}}\left[\|r^f(s_{t+1}, \widetilde{s}_{t+1}) - V^f(s_t, g)\|_2\right]. \quad (11)$$

Since $V^f(s_t, g)$ regresses towards a one-step reward instead of a cumulative return, the corresponding advantage function A^f can be calculated simply as $r^f(s_{t+1}, \widetilde{s}_{t+1}) - V^f(s_t, g)$. Finally, A^f will be used for low-level policy learning which will be further explained in Section 4.3.

4.3. Multiscale SCP for Comprehensive Guidance

Remarkably, through the intrinsic reward r^f defined above, the fast SCP not only provides pessimistic regularization but also serves as a one-step target for π^l to reach. An ideal π^f is able to guide π^l step by step, and maximizing r^f is sufficient. However, in practice, the predictions of next states are noisy. Planning states at a moderate distance, on the other hand, can offer enhanced long-term planning ability (Park et al., 2023). The empirical results in Section 5.3 also shows that a slow SCP tends to focus on long-term objectives such as navigation, while a fast SCP places more emphasis on short-term objectives such as locomotion. Consequently, our MSCP algorithm trains a slow h-SCP π^s ($h > 1$) in addition to the fast SCP π^f, utilizing both SCPs to provide comprehensive guidance for policy learning. For the slow SCP, the L2 distance of two states s_t, s_{t+h}, where h is large, is not useful to construct an intrinsic reward. Therefore, we adopt the hierarchical policy extraction structure following HIQL (Park et al., 2023). Specifically, π^s predicts subgoals that are h steps away and the policy π^l and advantage $\widetilde{A}^l$ are conditioned on these subgoals to utilize the planning ability of π^s (see Equation 7).

Finally, MSCP trains π^l with AWR-style objective, utilizing the guidance from multiscale SCPs:

$$\mathcal{J}_{\mathcal{B}}^l(\theta_{\pi^l}) = \frac{1}{|\mathcal{B}|}\sum_{\mathcal{B}}\Big[\exp\Big(\beta\Big(\widetilde{A}^l(s_t, s_{t+1}, s_{t+h}) + C^f A^f(s_t, s_{t+1}, g)\Big)\Big)\log\pi^l(a_t|s_t, s_{t+h})\Big] \quad (12)$$

where $A^f(s_t, s_{t+1}, g) = r^f(s_{t+1}, \widetilde{s}_{t+1}) - V^f(s_t, g)$ represents immediate guidance and pessimistic regularization from fast SCP, $\widetilde{A}^l(s_t, s_{t+1}, s_{t+h}) = V(s_{t+1}, s_{t+h}) -$

Algorithm 2 MSCP: Online Reinforcement Learning

Input: pretrained value functions and SCPs: V, $\bar{V}$, π^f, π^s, action-free dataset $\mathcal{D}$, horizon h, learning rate α, target upadte rate ϵ, number of steps N

Initialize low-level policy π^l with parameters θ_{π^l}
Initialize value function V^f with parameters θ_{V^f}
Initialize an empty replay buffer $\mathcal{R}$
for $i = 1$ **to** N **do**
  $(s_t, a_t, s_{t+1}, g) \leftarrow \text{rollout}(\pi^s, \pi^l)$
  Add the collected transition into $\mathcal{R}$
  Sample a batch $\mathcal{B}^{\mathcal{D}}$ of $(s_t, s_{t+1}, s_{t+k}, g)$ from $\mathcal{D}$
  Sample a batch $\mathcal{B}^{\mathcal{R}}$ of $(s_t, s_{t+1}, s_{t+k}, g)$ from $\mathcal{R}$
  $\mathcal{B} \leftarrow \mathcal{B}^{\mathcal{D}} \cup \mathcal{B}^{\mathcal{R}}$
  $\theta_V \leftarrow \theta_V - \alpha \nabla \mathcal{L}_{\mathcal{B}}^{V}(\theta_V)$ {Equation 5}
  $\theta_{V^f} \leftarrow \theta_{V^f} - \alpha \nabla \mathcal{L}_{\mathcal{B}}^{V^f}(\theta_{V^f})$ {Equation 11}
  $\theta_{\pi^l} \leftarrow \theta_{\pi^l} + \alpha \nabla \mathcal{J}_{\mathcal{B}^{\mathcal{R}}}^{l}(\theta_{\pi^l})$ {Equation 12}
  $\bar{\theta}_V \leftarrow \epsilon \theta_V + (1-\epsilon)\bar{\theta}_V$
  if also tune state-centric planners **then**
    $\theta_{\pi^f} \leftarrow \theta_{\pi^f} + \alpha \nabla \mathcal{J}_{\mathcal{B}}^{(1)}(\theta_{\pi^f})$ {Equation 6, $k=1$}
    $\theta_{\pi^s} \leftarrow \theta_{\pi^s} + \alpha \nabla \mathcal{J}_{\mathcal{B}}^{(h)}(\theta_{\pi^s})$ {Equation 6, $k=h$}
  end if
end for
return π^l, π^s

$V(s_t, s_{t+h})$ represents the slow SCP's guidance that focus more on long-term planning abilities, and C^f is a hyperparameter used to balance the supervision from the fast SCP and the slow SCP. The original r and γ in calculating $\widetilde{A}^l$ are absorbed into β as explained in Section 2.2.

4.4. Algorithm Summary

The pretraining and online stages of MSCP are presented in Algorithm 3 and Algorithm 2 respectively. In online RL, MSCP also finetunes the pretrained V with both online and offline data, and optionally tunes the two SCPs. This technique is commonly employed in offline-to-online RL to address the distributional shift problem and prevent catastrophic forgetting. The same trick is also applied to HIQL baseline for fairness. In tasks involving visual inputs, where predicting states in the image space can be difficult, we simply train a goal representation when learning the value function (Park et al., 2023). Both the planning of SCPs and the calculation of intrinsic rewards are conducted in this latent space of representation. Additional implementation details and hyperparameters can be found in Appendix B.

5. Experiment

We conduct extensive experiments in four challenging environments that prioritize long-term planning: the *AntMaze* and *Kitchen* in D4RL (Fu et al., 2020), *CALVIN* (Mees et al., 2022), and *Procgen* Maze (Cobbe et al., 2020). The environments are shown in Figure 5. We follow Park et al. (2023) for the setup of environments and datasets. We remove the action and reward labels from all the datasets and retain only state sequences. Throughout the experiments, we compare with ICVF (Ghosh et al., 2023), HIQL (Park et al., 2023), and AF-Guide (Zhu et al., 2023). We also include the results of an offline RL oracle that has access to the action labels of the dataset. We select HIQL as the oracle due to its performance which surpasses other baselines such as IQL (Kostrikov et al., 2022) and trajectory transformer (Janner et al., 2021). To demonstrate the difficulty of the tasks and the advantages of pretraining, we also include the results of online RL using SAC, which does not utilize any offline data. In all our experiments, we use 3M online environment steps and aggregate the results from five random seeds. Please refer to Appendix C for additional empirical results, including the impact of hyperparameters in Appendix C.2.

5.1. Environment Setup and Result

The *AntMaze*, *Kitchen*, and *CALVIN* environments present challenges in terms of long-term planning and learning navigation or manipulation skills from a continuous action space through pure online exploration. Additional details can be found in Appendix B.1. In Table 1, we display the normalized score of 11 tasks in these environments. ICVF's pretrained representation fails to benefit online learning. While AF-Guide successfully solves the easiest *umaze* tasks, its performance deteriorates rapidly as the difficulty increases. It shows that only training a one-step planner is insufficient to handle long-horizon planning. Our method MSCP demonstrates consistent strong performance in all the tasks. In Figure 2, we show the learning curves in four Antaze tasks, whose difficulties increase from left to right. The advantage of MSCP becomes increasingly evident as difficulty increases, showcasing MSCP's superiority in solving long-term planning problems. Remarkly, the pure online RL fails to solve any *AntMaze* tasks, which underscores the benefits of pretraining.

The training datasets for *Procgen-Maze-500* and *Procgen-Maze-1000* consist of 500 and 1000 mazes respectively, with visual input. A trained policy is also evaluated on held-out test mazes. The results in Table 2 show that our method surpasses the HIQL baseline, particularly in *Procgen-Maze-500* where the data distribution is narrower. This suggests that MSCP is effective in extracting valuable information from a dataset with limited coverage.

5.2. Effectiveness of Pessimistic Regularization

We conduct ablation study to show the existence of distributional shift from offline to online, and demonstrate that MSCP successfully mitigates the issue through the pes-

Table 1. Experiment results in eight *AntMaze* tasks, two *Kitchen* tasks, and one *CALVIN* task. We report the mean and standard deviation of the normalized scores across five random seeds.

TASK	OFFLINE RL ORACLE	PASSIVE RL ICVF	PASSIVE RL AF-GUIDE	PASSIVE RL HIQL	PASSIVE RL **MSCP (OURS)**	ONLINE RL (SAC)
ANTMAZE-UMAZE	85.4 ± 7.5	0.0 ± 0.0	**100.0 ± 0.0**	94.6 ± 2.1	97.3 ± 2.6	0.0 ± 0.0
ANTMAZE-UMAZE-DIVERSE	87.8 ± 1.1	0.0 ± 0.0	**98.0 ± 4.5**	91.9 ± 4.4	96.9 ± 2.6	0.0 ± 0.0
ANTMAZE-MEDIUM-PLAY	88.5 ± 5.1	0.0 ± 0.0	0.0 ± 0.0	69.6 ± 9.7	**96.5 ± 3.2**	0.0 ± 0.0
ANTMAZE-MEDIUM-DIVERSE	85.3 ± 6.2	0.0 ± 0.0	0.0 ± 0.0	81.9 ± 5.9	**92.7 ± 4.8**	0.0 ± 0.0
ANTMAZE-LARGE-PLAY	80.4 ± 4.4	0.0 ± 0.0	0.0 ± 0.0	57.3 ± 39.1	**92.3 ± 2.7**	0.0 ± 0.0
ANTMAZE-LARGE-DIVERSE	86.5 ± 6.5	0.0 ± 0.0	0.0 ± 0.0	49.6 ± 30.4	**94.2 ± 3.0**	0.0 ± 0.0
ANTMAZE-ULTRA-PLAY	44.6 ± 13.4	0.0 ± 0.0	0.0 ± 0.0	15.0 ± 14.5	**55.4 ± 18.0**	0.0 ± 0.0
ANTMAZE-ULTRA-DIVERSE	45.4 ± 15.5	0.0 ± 0.0	0.0 ± 0.0	26.9 ± 17.4	**80.8 ± 10.3**	0.0 ± 0.0
KITCHEN-PARTIAL	36.2 ± 20.7	0.0 ± 0.0	0.0 ± 0.0	13.1 ± 10.4	**33.0 ± 18.6**	17.0 ± 16.1
KITCHEN-MIXED	41.4 ± 21.5	0.0 ± 0.0	0.0 ± 0.0	14.8 ± 11.9	**39.8 ± 8.4**	10.0 ± 13.7
CALVIN	81.2 ± 15.2	44.9 ± 20.7	10.0 ± 13.7	44.5 ± 21.3	**65.0 ± 13.7**	50.0 ± 35.4

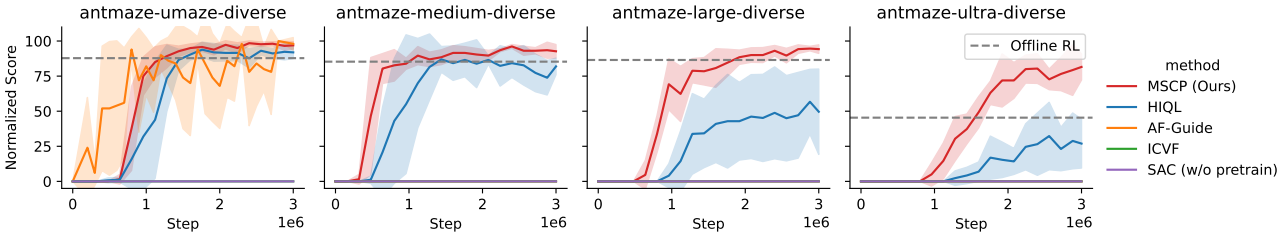

Figure 2. Learning curves in four *AntMaze* tasks: *umaze-diverse*, *medium-diverse*, *large-diverse*, and *ultra-diverse*. From left to right, the task difficulty quickly increases as the maze size grows. The x-axis represents environment steps, and the y-axis is the normalized score. The shaded area represents the standard deviation. The grey dashed line represents the performance of offline RL.

Table 2. Normalized scores in both the training and testing mazes of *Procgen-Maze-500* and *Procgen-Maze-1000* tasks.

TASK	OFFLINE RL ORACLE	PASSIVE RL HIQL	PASSIVE RL **MSCP(OURS)**
500-TRAIN	83.3 ± 10.1	60.4 ± 12.1	**77.7 ± 8.3**
500-TEST	70.7 ± 6.1	48.1 ± 8.7	**62.3 ± 9.3**
1000-TRAIN	89.3 ± 11.6	82.3 ± 4.2	**88.9 ± 5.8**
1000-TEST	86.0 ± 2.0	70.0 ± 7.4	**74.2 ± 7.8**

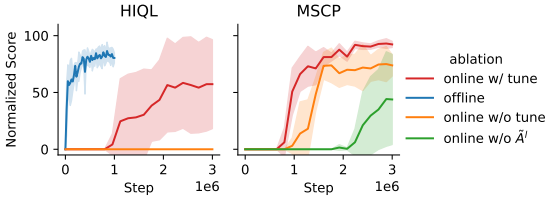

Figure 3. Ablation study for pessimistic regularization. The experiments are conducted in the *antmaze-large-diverse* task.

simistic regularization forced by the fast SCP. Recall that both MSCP and pervious method HIQL pretrain V to provide training signals for online policy. MSCP additionally has pessimistic guidance from a fast SCP. In the left part of Figure 3, HIQL fails if we fix the pretrained V and let the policy learn from online data (*online w/o tune*). However, when provided with an action-annotated version of the same dataset used in pretraining, HIQL learns quickly (*offline*). It shows that the pretrained V only provides useful guidance within the range of the dataset, and generalizes poorly on online data. If we continue to tune V with both offline and online data in online RL (*online w/ tune*), HIQL starts to learn useful behavior, which further proves the existence of distributional shift. For MSCP, we also consider the *online w/o* $\widetilde{A}^l$ case where we only use the one-step guidance from the fast SCP to train the policy (removing $\widetilde{A}^l$ in Equation 12). In the right part of Figure 3, when the pretrained V is fixed, MSCP still shows good performance, and is better than not using V at all (*online w/o* $\widetilde{A}^l$). It shows that the pretrained V still contributes to policy learning, indicating that the pessimistic regularization constrains the policy to stay close to the dataset, mitigating the extrapolation error and better leveraging the pretrained V. In Appendix C.3, we also visualize that, compared to HIQL, the policy learned by MSCP is closer to the policy learned offline.

5.3. Effectiveness of Multiscale Planning

We consider three ablation studies to evaluate the efficacy of multiscale planning: (1) the removal of fast SCP, which is HIQL; (2) the removal of $\widetilde{A}^l$ in policy loss as has been explained previously; and (3) the removal of the whole slow

Table 3. Ablation studies of the fast SCP, the guidance extracted by the slow SCP from the pretrained value function, and the whole slow SCP. All the components are critical for the effectiveness of multiscale planning.

TASK	**MSCP (OURS)**	W/O FAST SCP	W/O $\widetilde{A}^l$	W/O SLOW SCP
ANTMAZE-UMAZE-DIVERSE	**96.9** $\pm$ **2.6**	91.9 $\pm$ 4.4	48.8 $\pm$ 31.7	0.0 $\pm$ 0.0
ANTMAZE-LARGE-DIVERSE	**94.2** $\pm$ **3.0**	49.6 $\pm$ 30.4	40.0 $\pm$ 32.3	0.0 $\pm$ 0.0
ANTMAZE-ULTRA-DIVERSE	**80.8** $\pm$ **10.3**	26.9 $\pm$ 17.4	0.0 $\pm$ 0.0	0.0 $\pm$ 0.0
CALVIN	**65.0** $\pm$ **13.7**	44.5 $\pm$ 21.3	18.6 $\pm$ 32.5	0.0 $\pm$ 0.0

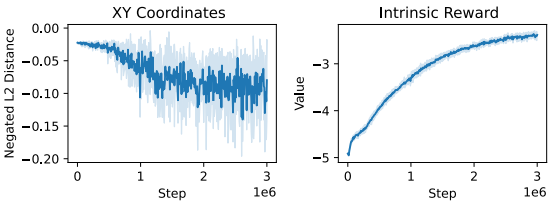

Figure 4. The intrinsic reward $r^f(s_{t+1}, \widetilde{s}_{t+1})$ and the negated L2 distance between the xy coordinates of s_{t+1} and $\widetilde{s}_{t+1}$.

SCP. In both (2) and (3), the policy only learns from the guidance of fast SCP. In (2), the policy still conditions on a closer subgoal predicted by slow SCP, while in (3), the policy directly conditions on ultimate goal g. As shown in Table 3, both the slow and fast SCPs have significant contributions to the overall efficiency of our MSCP algorithm.

We use *AntMaze* to provide an intuitive understanding of what objectives the two SCPs prioritize respectively. The state s in *AntMaze* consists of navigation features (xy coordinates of the ant) and locomotion features (relative positions and velocities of the robot's joints). Figure 4 shows that policy learns to maximize the intrinsic reward $r^f(s_{t+1}, \widetilde{s}_{t+1}) = -\|s_{t+1} - \widetilde{s}_{t+1}\|_2$. However, the distance between the xy coordinations of the actual next state s_{t+1} and $\widetilde{s}_{t+1}$ predicted by the fast SCP actually increases as training proceeds. It indicates that the fast SCP prioritizes providing guidance on locomotion, while the predicted next state's xy position can be too noisy to assist navigation. On the contrary, we visualize that the slow SCP plans appropriate subgoals to guide the navigation in Appendix C.4.

6. Conclusion

We demonstrate the provable statistical benefit of pretraining a goal-conditioned value function in passive RL. Building upon the insights, we propose a novel algorithm named Multiscale State-Centric Planners (MSCP) that leverages two state-centric planners to extract comprehensive guidance from imperfectly pretrained value functions. MSCP also forces pessimistic regularization to mitigate the extrapolation error arising from the distributional shift from offline to online. Empirical studies demonstrate the superiority of MSCP and confirm the significance of all components.

Acknowledgements

The authors would like to thank the anonymous reviewers for their insightful discussions and helpful suggestions.

Impact Statement

Our research primarily addresses the core challenges in passive reinforcement learning, without currently delving into potential societal ramifications that warrant explicit discussion in this statement.

References

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24:2312–2320, 2011.

Agarwal, R., Schuurmans, D., and Norouzi, M. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, pp. 104–114. PMLR, 2020.

Ajay, A., Kumar, A., Agrawal, P., Levine, S., and Nachum, O. Opal: Offline primitive discovery for accelerating offline reinforcement learning. *arXiv preprint arXiv:2010.13611*, 2020.

Andrychowicz, M., Wolski, F., Ray, A., Schneider, J., Fong, R., Welinder, P., McGrew, B., Tobin, J., Pieter Abbeel, O., and Zaremba, W. Hindsight experience replay. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Bacon, P.-L., Harb, J., and Precup, D. The option-critic architecture. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), Feb. 2017. doi: 10.1609/aaai.v31i1.10916. URL https://ojs.aaai.org/index.php/AAAI/article/view/10916.

Baker, B., Akkaya, I., Zhokov, P., Huizinga, J., Tang, J., Ecoffet, A., Houghton, B., Sampedro, R., and Clune, J. Video pretraining (vpt): Learning to act by watching unlabeled online videos. In Koyejo, S., Mohamed, S.,

Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 24639–24654. Curran Associates, Inc., 2022.

Ball, P. J., Smith, L., Kostrikov, I., and Levine, S. Efficient online reinforcement learning with offline data. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 1577–1594. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/ball23a.html.

Bauer, J., Baumli, K., Behbahani, F., Bhoopchand, A., Bradley-Schmieg, N., Chang, M., Clay, N., Collister, A., Dasagi, V., Gonzalez, L., Gregor, K., Hughes, E., Kashem, S., Loks-Thompson, M., Openshaw, H., Parker-Holder, J., Pathak, S., Perez-Nieves, N., Rakicevic, N., Rocktäschel, T., Schroecker, Y., Singh, S., Sygnowski, J., Tuyls, K., York, S., Zacherl, A., and Zhang, L. M. Human-timescale adaptation in an open-ended task space. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 1887–1935. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/bauer23a.html.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020.

Chebotar, Y., Vuong, Q., Hausman, K., Xia, F., Lu, Y., Irpan, A., Kumar, A., Yu, T., Herzog, A., Pertsch, K., Gopalakrishnan, K., Ibarz, J., Nachum, O., Sontakke, S. A., Salazar, G., Tran, H. T., Peralta, J., Tan, C., Manjunath, D., Singh, J., Zitkovich, B., Jackson, T., Rao, K., Finn, C., and Levine, S. Q-transformer: Scalable offline reinforcement learning via autoregressive q-functions. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=0I3su3mkuL.

Cobbe, K., Hesse, C., Hilton, J., and Schulman, J. Leveraging procedural generation to benchmark reinforcement learning. In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 2048–2056. PMLR, 13–18 Jul 2020. URL https://proceedings.mlr.press/v119/cobbe20a.html.

Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., Legg, S., and Kavukcuoglu, K. IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In Dy, J. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1407–1416. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/espeholt18a.html.

Fawzi, A., Balog, M., Huang, A., Hubert, T., Romera-Paredes, B., Barekatain, M., Novikov, A., Ruiz, F. J. R., Schrittwieser, J., Swirszcz, G., Silver, D., Hassabis, D., and Kohli, P. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930): 47–53, 2022. doi: 10.1038/s41586-022-05172-4.

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4RL: datasets for deep data-driven reinforcement learning. *CoRR*, abs/2004.07219, 2020. URL https://arxiv.org/abs/2004.07219.

Fujimoto, S., Meger, D., and Precup, D. Off-Policy Deep Reinforcement Learning without Exploration. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2052–2062. PMLR, June 2019. URL https://proceedings.mlr.press/v97/fujimoto19a.html.

Gangwani, T. and Peng, J. State-only imitation with transition dynamics mismatch. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HJgLLyrYwB.

Ghosh, D., Bhateja, C. A., and Levine, S. Reinforcement learning from passive data via latent intentions. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 11321–11339. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/ghosh23a.html.

Gupta, A., Kumar, V., Lynch, C., Levine, S., and Hausman, K. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. In Kaelbling, L. P., Kragic, D., and Sugiura, K.

(eds.), *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1025–1037. PMLR, 30 Oct–01 Nov 2020. URL https://proceedings.mlr.press/v100/gupta20a.html.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. B. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 15979–15988. IEEE, 2022. doi: 10.1109/CVPR52688.2022.01553. URL https://doi.org/10.1109/CVPR52688.2022.01553.

Hu, H., Yang, Y., Zhao, Q., and Zhang, C. On the role of discount factor in offline reinforcement learning. In *International Conference on Machine Learning*, pp. 9072–9098. PMLR, 2022.

Hu, H., Yang, Y., Zhao, Q., and Zhang, C. The provable benefits of unsupervised data sharing for offline reinforcement learning. *arXiv preprint arXiv:2302.13493*, 2023.

Hu, H., Yang, Y., Ye, J., Mai, Z., and Zhang, C. Unsupervised behavior extraction via random intent priors. *Advances in Neural Information Processing Systems*, 36, 2024.

Janner, M., Li, Q., and Levine, S. Offline reinforcement learning as one big sequence modeling problem. In Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 1273–1286. Curran Associates, Inc., 2021.

Jiang, Z., Zhang, T., Janner, M., Li, Y., Rocktäschel, T., Grefenstette, E., and Tian, Y. Efficient planning in a compact latent action space. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=cA77NrVEuqn.

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pp. 2137–2143. PMLR, 2020.

Jin, Y., Yang, Z., and Wang, Z. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, pp. 5084–5096. PMLR, 2021.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021. doi: 10.1038/s41586-021-03819-2.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

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

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 1179–1191, 2020.

Kumar, A., Agarwal, R., Geng, X., Tucker, G., and Levine, S. Offline q-learning on diverse multi-task data both scales and generalizes. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=4-k7kUavAj.

Lee, S., Seo, Y., Lee, K., Abbeel, P., and Shin, J. Offline-to-online reinforcement learning via balanced replay and pessimistic q-ensemble. In *Conference on Robot Learning*, pp. 1702–1712. PMLR, 2022.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. *CoRR*, abs/2005.01643, 2020. URL https://arxiv.org/abs/2005.01643. arXiv: 2005.01643.

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

Liu, F., Ling, Z., Mu, T., and Su, H. State alignment-based imitation learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rylrdxHFDr.

Ma, X., Yang, Y., Hu, H., Liu, Q., Yang, J., Zhang, C., Zhao, Q., and Liang, B. Offline reinforcement learning with value-based episodic memory. *arXiv preprint arXiv:2110.09796*, 2021.

Mees, O., Hermann, L., Rosete-Beas, E., and Burgard, W. CALVIN: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics Autom. Lett.*, 7(3):7327–7334, 2022. doi: 10.1109/LRA.2022.3180108. URL https://doi.org/10.1109/LRA.2022.3180108.

Nachum, O., Gu, S. S., Lee, H., and Levine, S. Data-efficient hierarchical reinforcement learning. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/e6384711491713d29bc63fc5eeb5ba4f-Paper.pdf.

Nair, A., Dalal, M., Gupta, A., and Levine, S. Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Nakamoto, M., Zhai, Y., Singh, A., Mark, M. S., Ma, Y., Finn, C., Kumar, A., and Levine, S. Cal-ql: Calibrated offline rl pre-training for efficient online fine-tuning. *arXiv preprint arXiv:2303.05479*, 2023.

Park, S., Ghosh, D., Eysenbach, B., and Levine, S. HIQL: Offline goal-conditioned RL with latent states as actions. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=cLQCCtVDuW.

Park, S., Kreiman, T., and Levine, S. Foundation policies with hilbert representations. *arXiv preprint arXiv:2402.15567*, 2024.

Peng, X. B., Kumar, A., Zhang, G., and Levine, S. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Prudencio, R. F., Máximo, M. R. O. A., and Colombini, E. L. A Survey on Offline Reinforcement Learning: Taxonomy, Review, and Open Problems. *CoRR*, abs/2203.01387, 2022. doi: 10.48550/arXiv.2203.01387. URL https://doi.org/10.48550/arXiv.2203.01387. arXiv: 2203.01387.

Radosavovic, I., Wang, X., Pinto, L., and Malik, J. State-only imitation learning for dexterous manipulation. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7865–7871, 2021. doi: 10.1109/IROS51168.2021.9636557.

Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-maron, G., Giménez, M., Sulsky, Y., Kay, J., Springenberg, J. T., Eccles, T., Bruce, J., Razavi, A., Edwards, A., Heess, N., Chen, Y., Hadsell, R., Vinyals, O., Bordbar, M., and de Freitas, N. A generalist agent. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=1ikK0kHjvj. Featured Certification, Outstanding Certification.

Schaul, T., Horgan, D., Gregor, K., and Silver, D. Universal value function approximators. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1312–1320, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/schaul15.html.

Seo, Y., Lee, K., James, S. L., and Abbeel, P. Reinforcement learning with action-free pre-training from videos. In Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., and Sabato, S. (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 19561–19579. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/seo22a.html.

Shi, L. X., Lim, J. J., and Lee, Y. Skill-based model-based reinforcement learning. In *6th Annual Conference on Robot Learning*, 2022. URL https://openreview.net/forum?id=iVxy2eO601U.

Siegel, N. Y., Springenberg, J. T., Berkenkamp, F., Abdolmaleki, A., Neunert, M., Lampe, T., Hafner, R., Heess, N., and Riedmiller, M. Keep doing what worked: Behavioral modelling priors for offline reinforcement learning. *arXiv preprint arXiv:2002.08396*, 2020.

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

van der Maaten, L. and Hinton, G. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

Vezhnevets, A. S., Osindero, S., Schaul, T., Heess, N., Jaderberg, M., Silver, D., and Kavukcuoglu, K. FeUdal networks for hierarchical reinforcement learning. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3540–3549. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/vezhnevets17a.html.

Wu, C., Tang, P., Yang, J., Hu, Y., Lv, T., Fan, C., and Zhang, C. Conservative offline policy adaptation in multi-agent games. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a. URL https://openreview.net/forum?id=C8pvL8Qbfa.

Wu, J., Ma, H., Deng, C., and Long, M. Pre-training contextualized world models with in-the-wild videos

for reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b. URL https://openreview.net/forum?id=8GuEVzAUQS.

Wu, Y., Tucker, G., and Nachum, O. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

Xiong, W., Zhong, H., Shi, C., Shen, C., Wang, L., and Zhang, T. Nearly minimax optimal offline reinforcement learning with linear function approximation: Single-agent mdp and markov game. *arXiv preprint arXiv:2205.15512*, 2022.

Xu, H., Jiang, L., Jianxiong, L., and Zhan, X. A policy-guided imitation approach for offline reinforcement learning. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 4085–4098. Curran Associates, Inc., 2022.

Yang, L. and Wang, M. Sample-optimal parametric q-learning using linearly additive features. In *International Conference on Machine Learning*, pp. 6995–7004. PMLR, 2019.

Yang, S., Schuurmans, D., Abbeel, P., and Nachum, O. Dichotomy of control: Separating what you can control from what you cannot. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=DEGjDDV22pI.

Yang, Y., Ma, X., Li, C., Zheng, Z., Zhang, Q., Huang, G., Yang, J., and Zhao, Q. Believe what you see: Implicit constraint approach for offline multi-agent reinforcement learning. *arXiv preprint arXiv:2106.03400*, 2021.

Yang, Y., Hu, H., Li, W., Li, S., Yang, J., Zhao, Q., and Zhang, C. Flow to control: Offline reinforcement learning with lossless primitive discovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 10843–10851, 2023b.

Ye, W., Zhang, Y., Abbeel, P., and Gao, Y. Become a proficient player with limited data through watching pure videos. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Sy-o2N0hF4f.

Yu, T., Kumar, A., Chebotar, Y., Hausman, K., Finn, C., and Levine, S. How to leverage unlabeled data in offline reinforcement learning. In *International Conference on Machine Learning*, pp. 25611–25635. PMLR, 2022.

Zhang, H., Xu, W., and Yu, H. Policy expansion for bridging offline-to-online reinforcement learning. *arXiv preprint arXiv:2302.00935*, 2023.

Zhu, D., Wang, Y., Schmidhuber, J., and Elhoseiny, M. Guiding online reinforcement learning with action-free offline pretraining, 2023.

A. MSCP Algorithm

In Algorithm 3, we show the pretraining of MSCP from a passive dataset $\mathcal{D}$ that is action-free and reward-free. It learns a goal-conditioned value function $V(s, g)$, a fast SCP π^f that predicts immediate next next, and a slow SCP π^s that predicts states that are h steps away.

Algorithm 3 MSCP: Passive Pretraining

Input: action-free dataset $\mathcal{D}$, horizon h, learning rate α, target upadte rate ϵ

Initialize goal-conditioned value function V and its target network $\bar{V}$, fast SCP π^f, h-distance slow SCP π^s with parameters θ_V, $\bar{\theta}_V$, θ_{π^f}, θ_{π^s} respectively
$\bar{\theta}_V \leftarrow \theta_V$
while not converged **do**
  Sample a batch $\mathcal{B}$ of $(s_t, s_{t+1}, s_{t+k}, g)$ from $\mathcal{D}$.
  $\theta_V \leftarrow \theta_V - \alpha \nabla \mathcal{L}_{\mathcal{B}}^V(\theta_V)$ {Equation 5}
  $\theta_{\pi^f} \leftarrow \theta_{\pi^f} + \alpha \nabla \mathcal{J}_{\mathcal{B}}^{(1)}(\theta_{\pi^f})$ {Equation 6, $k = 1$}
  $\theta_{\pi^s} \leftarrow \theta_{\pi^s} + \alpha \nabla \mathcal{J}_{\mathcal{B}}^{(h)}(\theta_{\pi^s})$ {Equation 6, $k = h$}
  $\bar{\theta}_V \leftarrow \epsilon \theta_V + (1 - \epsilon)\bar{\theta}_V$
end while
return V, $\bar{V}$, π^f, π^s

B. Implementation Details

B.1. Environment Description

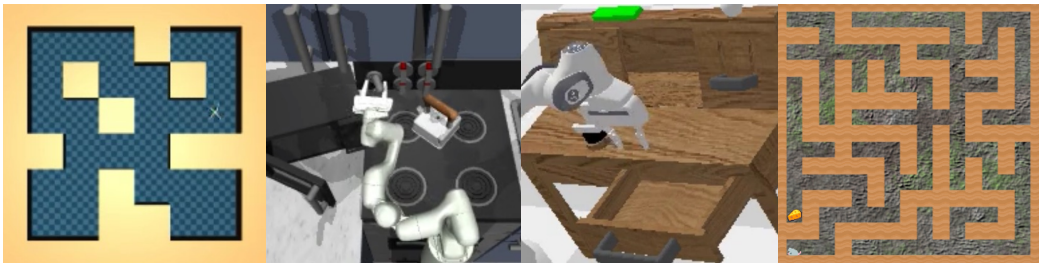

Figure 5. Environments used in experiments. From left to right: (1) AntMaze, (2) Kitchen, (3) CALVIN, (4) Procgen Maze.

We conduct extensive experiments in four challenging environments that prioritize long-term planning: the *AntMaze* and *Kitchen* environment in D4RL (Fu et al., 2020), *CALVIN* (Mees et al., 2022), and *Procgen* Maze (Cobbe et al., 2020). The environments are shown in Figure 5. We follow Park et al. (2023) for the setup of environments and datasets. We remove the action and reward labels from all datasets and retain only state sequences. For all environments, we report the normalized score which transforms the trajectory returns from $[0, R_{\text{max}}^{\text{env}}]$ into $[0, 100]$.

AntMaze *AntMaze* (Fu et al., 2020) is a navigation task where the policy controls an 8-DoF quadraped robot to walk from the starting position of the maze to the target position. The robot is only rewarded when it reaches the target. We use four maze layouts: *umaze*, *medium*, *large*, and *large*, whose difficulties gradually increase. We use the standard datasets provided by Fu et al. (2020); Jiang et al. (2023); Park et al. (2023)

Kitchen *Kitchen* (Fu et al., 2020; Gupta et al., 2020) is a manipulation task where the policy controls a 9-DoF robot arm to finish four subtasks in a kitchen environment, including opening the microwave, moving kettle, turning on light switch, and opening the sliding cabinet door. The robot receives a $+1$ upon the finishing of one subtask. We use the dataset provided by Fu et al. (2020).

CALVIN *CALVIN* (Mees et al., 2022) is similar to *Kitchen*, also featuring four manipulation subtasks. We use the dataset provided by Shi et al. (2022).

Procgen Maze *Procgen Maze* is part of a set of procedural generated environment (Cobbe et al., 2020) that focusing on evaluating generalization ability. It uses 64x64x3 image input instead of vector features. The policy controls with discrete actions. For *Procgen-Maze-500*, we use 500 mazes in both offline pretraining and online RL to learn. The performance of the policy is evaluated on both the training mazes and a set of held-out test mazes. It is in the same for *Procgen-Maze-1000*. We use the dataset provided by Park et al. (2023).

B.2. Hyperparameters

Our code is built upon Park et al. (2023). For all the tasks in *AntMaze* and *Kitchen*, we use the same set of hyperparameters, illustrated in Table 4. The hyperparameter C^f is newly introduced by MSCP, and we use 7.0 for all of these tasks. For *CALVIN*, listed in Table 5, we propose subgoals in a latent space of dimension 10, and use $C^f = 1$. We additionally tune both SCPs with both online and offline data in the online stage. In *Procgen Maze* (shown in table Table 6), in order to address the visual inputs, we also plans subgoals in the latent space. We set $C^f = 3$ and $h = 3$. We use the same network architecture as Park et al. (2023), using an IMPALA CNN (Espeholt et al., 2018) for processing visual inputs, and a 3 layer MLP with 512 hidden units for all other network. The goal representation is trained solely from the gradients of value updates. We employ no additional reconstruction loss or contrastive loss.

For fairness, we use the same set of hyperparameters for the HIQL baseline. For the offline oracle, we use HIQL to learn purely from the dataset that contains action labels. For the passive RL setting, we use HIQL to pretrain its value function and high-level planner on an action-free dataset, and then learns the low-level control policy with online collected data.

For both HIQL and MSCP, we also tune the pretrained value function in online RL, with half of the data sampled from the offline dataset, and the other half sampled from the replay buffer. HIQL cannot solve the *AntMaze* without this technique, while MSCP is also witnessed a slight decrease in performance (see Figure 3).

Table 4. Hyperparameters used in all *AntMaze* and *Kitchen* tasks.

EXPECTILE τ	0.7	EXPLORATION TEMPERATURE	1.0	DISCOUNT FACTOR γ	0.99
SLOW SCP h	25	TUNE VALUE	TRUE	TUNE SCPS	FALSE
NUM WORKERS	8	STEPS	3M	REPLAY BUFFER SIZE	300K
UPDATE RATIO	2	IQL TEMPERATURE β	1	BATCH SIZE	1024
C^f	7.0	LATENT GOAL SPACE	FALSE		

Table 5. Hyperparameters used in *CALVIN*.

EXPECTILE τ	0.7	EXPLORATION TEMPERATURE	1.0	DISCOUNT FACTOR γ	0.99
SLOW SCP h	25	TUNE VALUE	TRUE	TUNE SCPS	TRUE
NUM WORKERS	8	STEPS	3M	REPLAY BUFFER SIZE	300K
UPDATE RATIO	2	IQL TEMPERATURE β	1	BATCH SIZE	1024
C^f	1.0	LATENT GOAL SPACE	TRUE	LATENT DIMENSION	10

Table 6. Hyperparameters used in *Procgen Maze*.

EXPECTILE τ	0.7	EXPLORATION TEMPERATURE	1.0	DISCOUNT FACTOR γ	0.99
SLOW SCP h	3	TUNE VALUE	TRUE	TUNE SCPS	TRUE
NUM WORKERS	8	STEPS	3M	REPLAY BUFFER SIZE	300K
UPDATE RATIO	2	IQL TEMPERATURE β	1	BATCH SIZE	1024
C^f	3.0	LATENT GOAL SPACE	TRUE	LATENT DIMENSION	10

C. Additional Experiment Results

C.1. Learning Curves

We show the learning curves in all the experiment environments in Figure 6, Figure 7, and Figure 8. In *AntMaze*, we use 1M gradient steps in pretraining from passive dataset for our method MSCP and all other passive RL baselines. For *Kitchen*, *CALVIN*, and *Procgen Maze*, we use 500K gradient updates. In the online learning stage, all methods consume 3M environment steps. All results are aggregated with five random seeds.

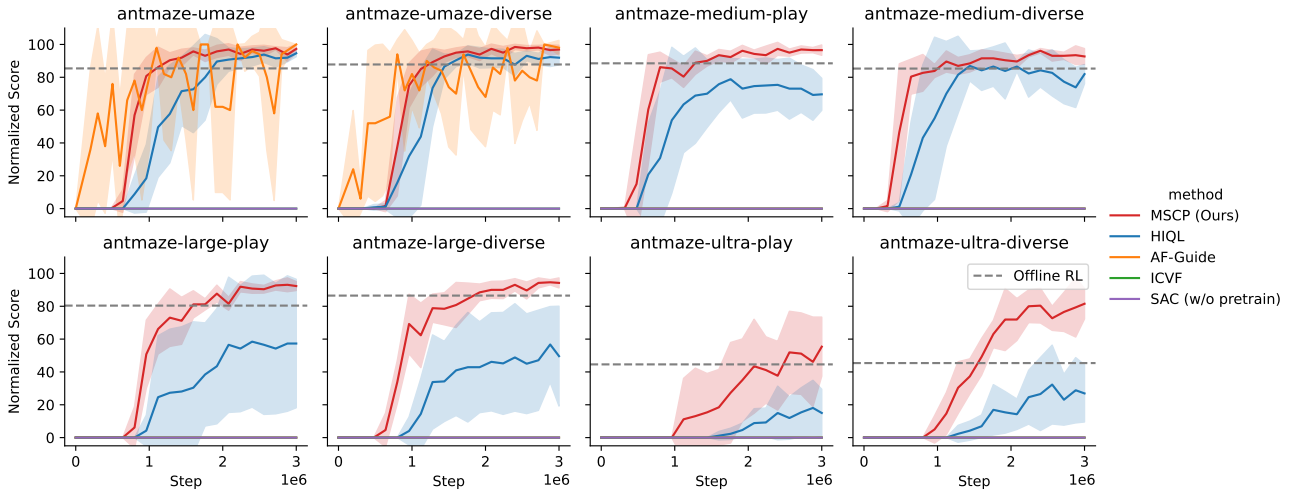

Figure 6. Learning curves in eight tasks of D4RL *AntMaze*, comparing our method MSCP with baselines. The x-axis is the environment step in online training, the y-axis represents normalized score. We report the mean and standard deviation of the normalized score across five random seeds.

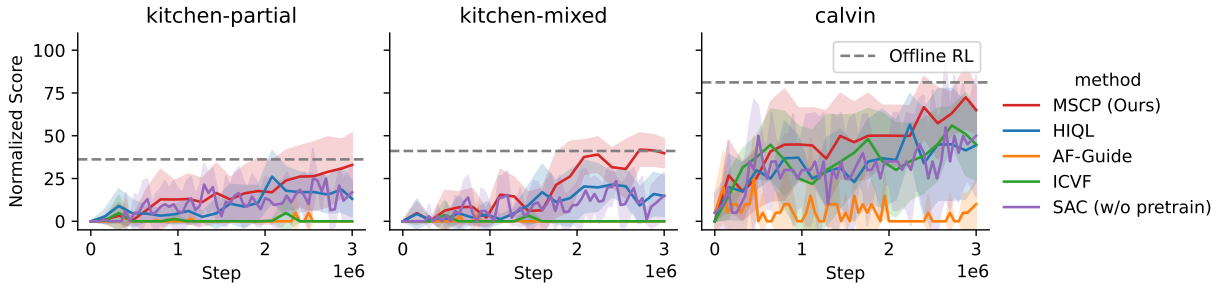

Figure 7. Learning curves in two tasks of *Kitchen*, and one task in *CALVIN*, comparing our method MSCP with baselines. The x-axis is the environment step in online training, the y-axis represents normalized score. We report the mean and standard deviation of the normalized score across five random seeds.

C.2. Hyperparameters

One major hyperparamter introduced by our method MSCP is $\mathcal{C}^f$, which balances the supervision from the fast SCP and the slow SCP as outlined in Equation 12. A larger $\mathcal{C}^f$ indicates a greater reliance on the fast SCP. In Table 7, we show the experiment results in the *antmaze-medium-diverse* and *antmaze-large-diverse* tasks with different $\mathcal{C}^f$ settings. The results reported in the paper use $\mathcal{C}^f = 7$, and $\mathcal{C}^f = 0$ is equivalent to the HIQL baseline. The "w/o $\widetilde{A}^l$" refers to the case where only the fast SCP is used (see Section 5.2 for details). The results show that MSCP is not sensitive to $\mathcal{C}^f$ as long as it lies within a suitable range.

Another hyperparameter is whether to tune the state-centric planners (SCPs) (Lines 14-16 of Algorithm 2). For the *AntMaze* and *Kitchen* tasks, we find that this choice does not have a significant impact. In contrast, as shown in Table 8, for the *CALVIN* and *Procgen-Maze* tasks, tuning the SCPs is necessary.

Lastly, we examine the impact of using different exploration temperatures in online fine-tuning. In the results reported in the paper, we set the exploration temperature to 1. As shown in Table 9, compared with $\mathcal{C}^f$, MSCP is more sensitive to the exploration temperature.

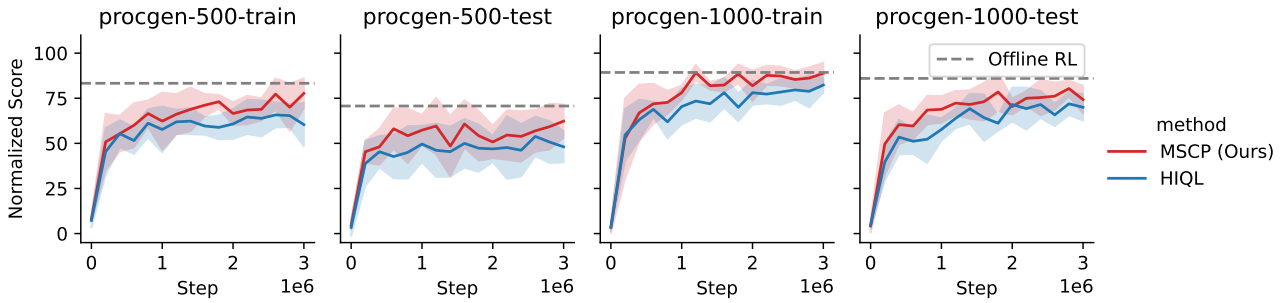

Figure 8. Learning curves in two Procgen Maze tasks. The *Procgen-500* consists of 500 mazes for training in both the offline dataset and online learning, while *Procgen-1000* uses 1000 maze. The evaluation results on both of the training mazes and a set of held-out testing mazes are reported. The x-axis is the environment step in online training, the y-axis represents normalized score. We report the mean and standard deviation of the normalized score across five random seeds.

Table 7. Experiments in the *antmaze-medium-diverse* and *antmaze-large-diverse* tasks with different $\mathcal{C}^f$ settings.

$\mathcal{C}^f$	ANTMAZE-MEDIUM-DIVERSE	ANTMAZE-LARGE-DIVERSE
0	81.9 ± 5.9	48.6 ± 29.7
1	85.1 ± 12.9	75.0 ± 5.1
3	94.2 ± 1.9	87.5 ± 9.5
5	90.4 ± 4.1	93.8 ± 3.2
7	92.7 ± 4.8	94.2 ± 3.0
10	95.6 ± 2.0	91.0 ± 5.6
20	91.3 ± 4.1	90.4 ± 6.9
w/o $\widetilde{A}^l$	69.2 ± 26.3	40.0 ± 32.3

C.3. Visualization of Learned Policies with t-SNE

We use t-SNE (van der Maaten & Hinton, 2008) to visualize the behavior of the policies learned by our method MSCP and baseline algorithm HIQL, compared against the policy learned fully from an offline dataset that reveals the action labels. We collect multiple trajectories with each of the policies in *AntMaze-Large-Diverse*, and embed all of the states encountered into a 2D space with t-SNE. In Figure 9, the backlight represents the state distribution of the offline trained policy. The red and yellow dots indicate the states visited by MSCP and HIQL respectively. As can be seen in the figure, most states visited by MSCP are also within the support of the offline policy. On the contrary, a large portion of states (marked out with a green ellipse) visited by HIQL lie outside the distribution of the offline policy. It indicates that, compared with HIQL, the policy learned by MSCP is much closer to the offline policy, showing the MSCP's ability to efficiently explore within the support of the dataset.

C.4. Visualization of Subgoals Planned by Slow SCP

In Figure 10, we demonstrate a sample trajectory collected by MSCP policy in the *AntMaze-Large-Diverse* task, where an ant robot needs to walk from the bottom left corner to the top right corner of the maze. The slow SCP predicts a state that is 25 steps away to serve as a subgoal to guide the ant. The red squares indicate the xy positions of the subgoals predicted by the slow SCP. The slow SCP is able to plan subgoals at an appropriate distance along the whole trajectory. It shows that the slow SCP can provide valuable guidance in navigation.

Table 8. Experiments of whether to tune the SCPs.

	W/ SCP FINETUNING	W/O SCP FINETUNING
CALVIN	65.0 ± 13.7	29.3 ± 28.5
PROCGENMAZE 500-TRAIN	77.7 ± 8.3	41.4 ± 0.04
PROCGENMAZE 500-TEST	62.3 ± 9.3	26.0 ± 0.04

Table 9. Experiments of exploration temperature.

EXPLORATION TEMPERATURE	0.5	0.8	0.9	1	1.5	2	5.0
ANTMAZE-MEDIUM-DIVERSE	0	84.0 ± 11.8	96.8 ± 1.1	92.7 ± 4.8	78.8 ± 15.3	44.7 ± 37.4	0
ANTMAZE-LARGE-DIVERSE	0	0	81.4 ± 10.6	94.2 ± 3.0	79.5 ± 4.8	0	0

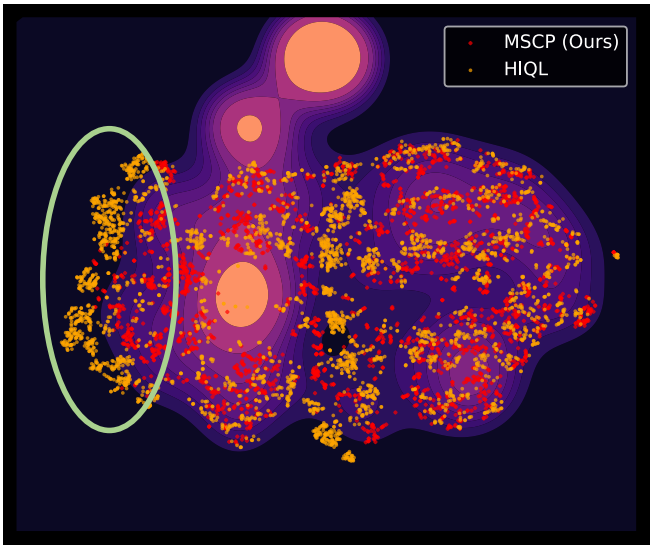

Figure 9. The t-SNE visualization of the state distributions of policies learned by MSCP and HIQL. The backlight represents the state distribution of an offline policy. The green ellipse mark out a portion of states visited by HIQL that lie outside the distribution of offline policy. The policy learned by MSCP behaves much similarly to an offline policy, indicating its ability to explore and learn within the range of the dataset.

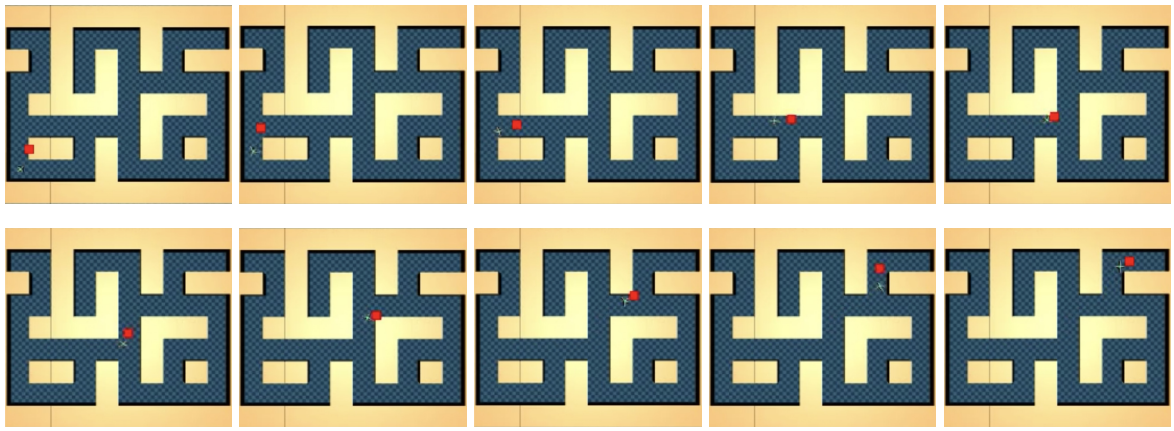

Figure 10. A sample trajctory of MSCP in *AntMaze-Large-Diverse*, where the red squres represent the subgoals predicted by the slow SCP. An ant robot needs to walk from the bottom left corner to the top right corner of the maze to finish the task.

D. Identifiable Dynamics

In this section, we given an explanation and some examples for the identifiable dynamics in Definition 3.1.

Intuitively, when the dynamics are identifiable, we know all the potential outcomes (i.e., distribution of the next state) of possible actions. This is a much weaker assumption than knowing the whole dynamics, since we may not know the exact mapping between the action and the outcome. One intuitive example is the deterministic dynamics, which are always identifiable with respect to reachable next states $\Phi(s) = \{s' \mid \exists\, a, P(s'|s,a) = 1\}$. While we don't know the exact mapping from the outcome s' and action a, we can still calculate the Bellman update by taking the maximum over all possible outcomes s'. Another example is Linear–quadratic–Gaussian (LQG) control with dynamics $x_{t+1} = Ax_t + Bu_t + v_t$ where $v_t \sim \mathcal{N}(0, 1)$. While we may not know the exact parameter for the dynamics, we know the distribution for the next state must be Gaussian. Then the LQG is identifiable with $\Phi(x) = \{\mathcal{N}(x_0, 1) \mid \exists u, x_0 = Ax + Bu\}$. From the above example, we can see that the identifiablity assumption is general than deterministic dynamics, as usually assumed in prior works (Park et al., 2023; Ghosh et al., 2023).

E. Details of Algorithms 1

Here we provide a detailed version of Algorithms 1, as shown in Algorithm 4.

Algorithm 4 Online Learning with Passive Value Iteration

Input: Passive dataset $\mathcal{D} = \{s_1, s_2, \ldots, s_N\}$, potential goals $\mathcal{G}$, parameter ζ

▷ Offline Phase
for every g in $\mathcal{G}$ **do**
  Initialization: Set $\widehat{V}(s, g) \leftarrow 0$
  Construct the uncertainty quantifier

$$\Gamma(s, s') \leftarrow \beta \cdot \left(\phi(s, s')^\top \Lambda_{\mathcal{D}}^{-1} \phi(s, s')\right)^{1/2}, \tag{13}$$

  where $\Lambda_{\mathcal{D}} = \sum_{i=1}^{N} \phi(s_i, s_{i+1}) \phi(s_i, s_{i+1})^\top$.
  while not converged **do**
    Compute the parameter for the state-state Q-value function via Ridge regression

$$\widehat{\nu}_t \leftarrow \operatorname*{argmin}_{\nu \in \mathbb{R}^d} \sum_{\tau=1}^{N} \left(R_\tau - \phi(s_\tau, s_{\tau+1})^\top \nu\right)^2 + \frac{\lambda}{2} \|\nu\|_2, \tag{14}$$

    Construct approximate operator

$$(\widehat{\mathbb{T}} V)(s, s') \leftarrow \phi(s, s', g)^\top \widehat{\nu}_t. \tag{15}$$

    Compute pessimistic Q-value function $\widehat{Q}(s, z, g)$

$$\widehat{Q}(s, z, g) \leftarrow \mathbb{E}_{s' \sim P_z(\cdot|s)} \left[(\widehat{\mathbb{T}} V)(s, s') - \Gamma(s, s', g)\right] \tag{16}$$

    Compute the pessimistic value function

$$\widehat{V}(\cdot, g) \leftarrow \max_{z \in \mathcal{Z}} \widehat{Q}(\cdot, z, g). \tag{17}$$

  end while
end for
▷ Online Phase
Randomly initialize π_1.
Receive the target goal g and initial state s_1.
for $t = 1, \ldots, T$ **do**
  Execute $a_t \sim \pi_t(\cdot|s_t)$.
  Receive reward $r(s_{t+1})$ and compute approximate advantage $\widehat{A}(s_t, a_t, g) = r(s_t, g) + \gamma \widehat{V}(s_{t+1}, g) - \widehat{V}(s_t, g)$.
  Estimate the parameter of the optimistic advantage function via Ridge regression

$$\widehat{w}_t \leftarrow \operatorname*{argmin}_{w \in \mathbb{R}^d} \sum_{i=1}^{t} (\widehat{A}(s_t, a_t, g) - \phi(s, a)^\top w)^2 + \frac{\lambda}{2} \|w\|_2 \tag{18}$$

  Compute the optimistic advantage function

$$\widehat{A}_t(s, a, g) \leftarrow \phi(s, a)^\top \widehat{w}_t + \beta_t (\phi(s, a)^\top \Sigma_t^{-1} \phi(s, a))^{1/2} \tag{19}$$

  Update policy

$$\pi_{t+1}(\cdot|s, g) \leftarrow \operatorname*{argmax}_{\pi} \langle \pi(\cdot|s, g), \widehat{A}_t(\cdot, s, g) \rangle_{\mathcal{A}}. \tag{20}$$

end for

F. Missing Proofs and Auxiliary Lemmas

F.1. Proof of Theorem 3.2

Proof. We prove a slightly weaker version of Theorem 3.2 and we aim to show that

$$|\widehat{V}(s,g) - V^\star(s,g)| \le \frac{2cr_{\max}}{(1-\gamma)^2}\sqrt{\frac{d^3\zeta}{NC^\dagger_{\mathcal{G}}}}$$

with probability $1-2\delta$. This error bound has a $d^{3/2}$ dependence on the dimension of the problem. However, this can be improved to d by using a fine-grained analysis as in Xiong et al. (2022).

Note that the following analysis is applicable to any goal g, we omit the parameter g for notational simplicity. For a sufficiently large λ, it is easy to see that $\mathcal{T}\widehat{V} := \max_z \mathbb{E}_z(\widehat{\mathbb{T}}\widehat{V} - \Gamma)$ is a contraction. Therefore Algorithm 1 converges and we have

$$\begin{aligned}
\widehat{V}(s) &= \max_z \widehat{Q}(s,z),\\
\widehat{Q}(s,s') &= \widehat{\mathbb{T}}\widehat{V} - \Gamma(s,s'),\\
\widehat{Q}(s,z) &= \mathbb{E}_{s'\sim P_z(\cdot|s)}\widehat{Q}(s,s').
\end{aligned}$$

Let

$$\delta(s,s') = \mathbb{T}\widehat{V}(s) - \widehat{Q}(s,s') = \mathbb{T}\widehat{V}(s) - \widehat{\mathbb{T}}\widehat{V} + \Gamma(s,s'). \tag{21}$$

Under the condition of Lemma F.2, it holds that

$$0 \le \delta(s,s') \le 2\Gamma(s,s'), \text{for all } s,s'. \tag{22}$$

Note that under the identifiability condition, taking maximum over z is the same as taking the maximum over a, then we have

$$\begin{aligned}
&V^\star(s) - \widehat{V}(s)\\
=&\mathbb{E}_{z\sim\pi^\star, s'\sim\mathcal{P}(\cdot|s,z)}\left[r(s,s') + \gamma V^\star(s')\right] - \mathbb{E}_{z\sim\widehat{\pi}, s'\sim\mathcal{P}(\cdot|s,z)}\left[\widehat{Q}(s,s')\right]\\
=&\mathbb{E}_{z\sim\pi^\star, s'\sim\mathcal{P}(\cdot|s,z)}\left[r(s,s') + \gamma V^\star(s') - \widehat{Q}(s,s')\right] + \mathbb{E}_{z\sim\pi^\star, s'\sim\mathcal{P}(\cdot|s,z)}\left[\widehat{Q}(s,s')\right] - \mathbb{E}_{z\sim\widehat{\pi}, s'\sim\mathcal{P}(\cdot|s,z)}\left[\widehat{Q}(s,s')\right]\\
=&\mathbb{E}_{z\sim\pi^\star, s'\sim\mathcal{P}(\cdot|s,z)}\left[r(s,s') + \gamma\widehat{V}(s') - \widehat{Q}(s,s')\right] + \gamma\mathbb{E}_{z\sim\pi^\star, s'\sim\mathcal{P}(\cdot|s,z)}\left[V^\star(s') - \widehat{V}(s')\right]\\
&+ \mathbb{E}_{z\sim\pi^\star, s'\sim\mathcal{P}(\cdot|s,z)}\left[\widehat{Q}(s,s')\right] - \mathbb{E}_{z\sim\widehat{\pi}, s'\sim\mathcal{P}(\cdot|s,z)}\left[\widehat{Q}(s,s')\right]\\
=&\mathbb{E}_{z\sim\pi^\star, s'\sim\mathcal{P}(\cdot|s,z)}\left[r(s,s') + \gamma\widehat{V}(s') - \widehat{Q}(s,s')\right] + \gamma\mathbb{E}_{z\sim\pi^\star, s'\sim\mathcal{P}(\cdot|s,z)}\left[V^\star(s') - \widehat{V}(s')\right]\\
&+ \left\langle\widehat{Q}(s,z), \pi^\star(z\,|\,s) - \widehat{\pi}(z\,|\,s)\right\rangle_{\mathcal{Z}}\\
=&\mathbb{E}_{z\sim\pi^\star, s'\sim\mathcal{P}(\cdot|s,z)}\left[\delta(s,s')\right] + \left\langle\widehat{Q}(s,z), \pi^\star(z\,|\,s) - \widehat{\pi}(z\,|\,s)\right\rangle_{\mathcal{Z}} + \cdots
\end{aligned} \tag{23}$$

$$\begin{aligned}
=&\mathbb{E}_{\pi^\star}\left[\sum_{t=0}^{\infty}\gamma^t\delta(s_t,s_{t+1})\,|\,s_0 = s\right] + \mathbb{E}_{\pi^\star}\left[\sum_{t=0}^{\infty}\gamma^t\left\langle\widehat{Q}(s_t,z), \pi^\star(z\,|\,s_t) - \widehat{\pi}(z\,|\,s_t)\right\rangle_{\mathcal{Z}}\,|\,s_0 = s\right]\\
\le&\mathbb{E}_{\pi^\star}\left[\sum_{t=0}^{\infty}\gamma^t\delta(s_t,s_{t+1})\,\middle|\,s_0 = s\right]\\
\le&2\mathbb{E}_{\pi^\star}\left[\sum_{t=0}^{\infty}\gamma^t\Gamma(s_t,s_{t+1})\,\middle|\,s_0 = s\right]\\
=&2\beta\mathbb{E}_{\pi^\star}\left[\sum_{t=0}^{\infty}\gamma^t\left(\phi(s_t,s_{t+1})^\top\Lambda^{-1}\phi(s_t,s_{t+1})\right)^{1/2}\,\middle|\,s_0 = s\right].
\end{aligned} \tag{24}$$

Here Equation 23 recursively expands $V^\star(s') - \widehat{V}(s')$. The first inequality follows from the fact that $\widehat{\pi}(\cdot|s) = \operatorname{argmax}_\pi \left\langle \widehat{Q}(s,z), \pi(z|s) \right\rangle_{\mathcal{Z}}$ and the second inequality follows from Equation 22.

Then the following event

$$\mathcal{E} = \left\{ V^\star(s) - \widehat{V}(s) \le 2\beta \mathbb{E}_{\pi^\star}\Big[\sum_{t=0}^{\infty} \gamma^t \big(\phi(s_t, s_{t+1})^\top \Lambda^{-1} \phi(s_t, s_{t+1})\big)^{1/2} \,\Big|\, s_0 = s \Big] \text{ for all } s \in \mathcal{S} \right\} \tag{25}$$

holds with probability $1 - \xi/2$. From the assumption in Equation 4, the following event

$$\mathcal{E}^\dagger = \left\{ C_{\mathcal{G}}^\dagger \cdot \frac{1}{N} \sum_{\tau=1}^{N} \phi(s_\tau, a_\tau)\phi(s_\tau, a_\tau)^\top \succeq \mathbb{E}_{\pi^\star}\left[\phi(s_t, s_{t+1})\phi(s_t, s_{t+1})^\top \,|\, s_0 = s \right] \text{ for all } s \in \mathcal{S} \right\}$$

also holds with probability $1 - \xi/2$. Then from the union bound, the event $\mathcal{E} \cap \mathcal{E}^\dagger$ holds with probability $1 - \xi$. We condition on this event here after. By the Cauchy-Schwarz inequality, we have

$$\begin{aligned}
&\mathbb{E}_{\pi^\star}\Big[\sum_{t=0}^{\infty} \gamma^t \big(\phi(s_t, s_{t+1})^\top \Lambda^{-1} \phi(s_t, s_{t+1})\big)^{1/2} \,\Big|\, s_0 = s \Big] \\
&\quad = \frac{1}{1-\gamma} \mathbb{E}_{d^{\pi^\star}}\left[\sqrt{\operatorname{Tr}\left(\phi(s,s')^\top \Lambda^{-1} \phi(s,s') \right)} \,\Big|\, s_0 = s \right] \\
&\quad = \frac{1}{1-\gamma} \mathbb{E}_{d^{\pi^\star}}\left[\sqrt{\operatorname{Tr}\left(\phi(s,s')\phi(s,s')^\top \Lambda^{-1} \right)} \,\Big|\, s_0 = s \right] \\
&\quad \le \frac{1}{1-\gamma} \sqrt{\operatorname{Tr}\left(\mathbb{E}_{d^{\pi^\star}}\left[\phi(s,s')\phi(s,s')^\top \,|\, s_0 = s \right] \Lambda^{-1} \right)} \\
&\quad = \frac{1}{1-\gamma} \sqrt{\operatorname{Tr}\left(\Sigma_{\pi^\star, s}^\top \Lambda^{-1} \right)},
\end{aligned} \tag{26}$$

for all $s \in \mathcal{S}$. On the event $\mathcal{E} \cap \mathcal{E}^\dagger$, we have

$$\begin{aligned}
V^\star(s) - \widehat{V}(s) &\le 2\beta \mathbb{E}_{\pi^\star}\Big[\sum_{t=0}^{\infty} \gamma^t \big(\phi(s_t, s_{t+1})^\top \Lambda^{-1} \phi(s_t, s_{t+1})\big)^{1/2} \,\Big|\, s_0 = s \Big] \\
&\le \frac{2\beta}{1-\gamma} \sqrt{\operatorname{Tr}\left(\Sigma_{\pi^\star, s} \cdot \big(I + \frac{1}{C_{\mathcal{G}}^\dagger} \cdot N \cdot \Sigma_{\pi^\star, s} \big)^{-1} \right)} \\
&= \frac{2\beta}{1-\gamma} \sqrt{\sum_{j=1}^{d} \frac{\lambda_j(s)}{1 + \frac{1}{C_{\mathcal{G}}^\dagger} \cdot N \cdot \lambda_j(s)}}.
\end{aligned}$$

Here $\{\lambda_j(s)\}_{j=1}^d$ are the eigenvalues of $\Sigma_{\pi^\star, s}$ for all $s \in \mathcal{S}$, the first inequality follows from the definition of $\mathcal{E}$ in Equation 25, and the second inequality follows from Equation 26 and the definition of $\mathcal{E}^\dagger$ in Equation 4. Meanwhile, by Definition 2.1, we have $\|\phi(s,s')\| \le 1$ for all $(s,s') \in \mathcal{S} \times \mathcal{S}$. By Jensen's inequality, we have

$$\|\Sigma_{\pi^\star, s}\|_{\text{op}} \le \mathbb{E}_{\pi^\star}\left[\|\phi(s,s')\phi(s,s')^\top\|_{\text{op}} \,|\, s_0 = s \right] \le 1 \tag{27}$$

for all $s \in \mathcal{S}$. As $\Sigma_{\pi^\star, s}$ is positive semidefinite, we have $\lambda_j(s) \in [0,1]$ for all $s \in \mathcal{S}$ and all $j \in [d]$. Hence, on $\mathcal{E} \cap \mathcal{E}^\dagger$, we have

$$\begin{aligned}
V^\star(s) - \widehat{V}(s) &\le \frac{2\beta}{1-\gamma} \sqrt{\sum_{j=1}^{d} \frac{\lambda_j(s)}{1 + \frac{1}{C_{\mathcal{G}}^\dagger} \cdot N \cdot \lambda_j(s)}} \\
&\le \frac{2\beta}{1-\gamma} \sqrt{\sum_{j=1}^{d} \frac{1}{1 + \frac{1}{C_{\mathcal{G}}^\dagger} \cdot N}} \le \frac{2c r_{\max}}{(1-\gamma)^2} \sqrt{C_{\mathcal{G}}^\dagger d^3 \zeta / N}
\end{aligned}$$

for all $x \in \mathcal{S}$, where the second inequality follows from the fact that $\lambda_j(s) \in [0,1]$ for all $s \in \mathcal{S}$ and all $j \in [d]$, while the third inequality follows from the choice of the scaling parameter $\beta > 0$. Combining the result in Lemma F.1, we have the conclusion in Theorem 3.2. □

Lemma F.1. *Under the event in Lemma F.2, we have*

$$\widehat{V}(s) - V^\star(s) \leq 0 \tag{28}$$

with probability $1-\delta$.

Proof. Note that $V^{\widehat{\pi}}(s) \leq V^\star(s)$, we only need to show that $\widehat{V}(s) - V^{\widehat{\pi}}(s) \leq 0$.

Let

$$\delta(s,s') = \mathbb{T}\widehat{V}(s,s') - \widehat{\mathbb{T}}\widehat{V}(s,s') = r(s,s') + \gamma\widehat{V}(s') - \widehat{\mathbb{T}}\widehat{V}(s,s'), \tag{29}$$

we have

$$\begin{aligned}
\widehat{V}(s) - V^{\widehat{\pi}}(s) =& \mathbb{E}_{z\sim\widehat{\pi}}\left[\widehat{Q}(s,s')\right] - \mathbb{E}_{z\sim\widehat{\pi}, s'\sim\mathcal{P}(\cdot|s,z)}\left[r(s,s') + \gamma V^{\widehat{\pi}}(s')\right] \\
=& \mathbb{E}_{z\sim\widehat{\pi}, s'\sim\mathcal{P}(\cdot|s,s')}\left[\widehat{Q}(s,s') - r(s,s') - \gamma\widehat{V}(s')\right] \\
&+ \gamma\mathbb{E}_{z\sim\widehat{\pi}, s'\sim\mathcal{P}(\cdot|s,z)}\left[\widehat{V}(s') - V^{\widehat{\pi}}(s')\right] \\
=& -\mathbb{E}_{\widehat{\pi}}\left[\delta(s,s')\right] + \gamma\mathbb{E}_{z\sim\widehat{\pi}, s'\sim\mathcal{P}(\cdot|s,z)}\left[\widehat{V}(s') - V^{\widehat{\pi}}(s')\right] \\
=& -\mathbb{E}_{\widehat{\pi}}\left[\delta(s,s')\right] + \cdots \\
=& -\mathbb{E}_{\widehat{\pi}}\left[\sum_{t=0}^{\infty}\gamma^t\delta(s_t,s_{t+1}) \,|\, s_0 = s\right].
\end{aligned}$$

Then under the condition of Lemma F.2, it holds that

$$0 \leq \delta(s,s') \leq 2\Gamma(s,s'), \text{for all } s,s', \tag{30}$$

Then we have the result immediately.

□

Lemma F.2 (ξ-Quantifiers)**.** *Let*

$$\lambda = 1, \quad \beta = c\cdot dV_{max}\sqrt{\zeta}, \quad \zeta = \log\left(2dN/(1-\gamma)\xi\right). \tag{31}$$

And Let Then $\Gamma(s,s') = \beta\cdot\left(\phi(s,s')^\top\Lambda^{-1}\phi(s,s')\right)^{1/2}$ *specified in Equation 13 satisfies that with probability at least* $1-\xi$*,*

$$|\mathbb{T}\widehat{V}(s,s') - \widehat{\mathbb{T}}\widehat{V}(s,s')| \leq \Gamma(s,s') = \beta\sqrt{\phi(s,s')^\top\Lambda^{-1}\phi(s,s')}, \forall (s,s') \in \mathcal{S}\times\mathcal{S}. \tag{32}$$

Proof. we have

$$\begin{aligned}
\mathbb{T}\widehat{V} - \widehat{\mathbb{T}}\widehat{V} &= \phi(s,s')^\top(\nu - \widehat{\nu}) \\
&= \phi(s,s')^\top\nu - \phi(s,s')\Lambda^{-1}\left(\sum_{\tau=1}^{N}\phi_\tau\cdot(r(s_{\tau+1}) + \gamma\widehat{V}(s_{\tau+1}))\right) \\
&= \underbrace{\phi(s,s')^\top\nu - \phi(s,s')\Lambda^{-1}\left(\sum_{\tau=1}^{N}\phi_\tau\phi_\tau^\top\nu\right)}_{\text{(i)}} + \underbrace{\phi(s,s')\Lambda^{-1}(\sum_{\tau=1}^{N}\phi_\tau\phi_\tau^\top\nu - \sum_{\tau=1}^{N}\phi_\tau(r_\tau + \gamma\widehat{V}(s_{\tau+1}))}_{\text{(ii)}},
\end{aligned} \tag{33}$$

Then we bound (i) and (ii), respectively.

For (i), we have

$$\begin{aligned}\text{(i)} &= \phi(s,s')^\top \nu - \phi(s,s')\Lambda^{-1}(\Lambda - \lambda I)\nu \\ &= \lambda \phi(s,s')\Lambda^{-1}\nu \\ &\le \lambda ||\phi(s,s')||_{\lambda^{-1}} ||\nu||_{\lambda^{-1}} \\ &\le V_{\max}\sqrt{d\lambda}\sqrt{\phi(s,s')^\top \Lambda^{-1}\phi(s,s')},\end{aligned} \tag{34}$$

where the first inequality follows from Cauchy-Schwartz inequality. The second inequality follows from the fact that $||\Lambda^{-1}||_{\text{op}} \le \lambda^{-1}$ and Lemma F.3.

For notation simplicity, let $\epsilon_\tau = r_\tau + \gamma \widehat{V}(s_{\tau+1}) - \phi_\tau^\top \nu$, then we have

$$\begin{aligned}|\text{(ii)}| &= \phi(s,s')\Lambda^{-1}\sum_{\tau=1}^N \phi_\tau \epsilon_\tau \\ &\le ||\sum_{\tau=1}^N \phi_\tau \epsilon_\tau||_{\Lambda^{-1}} \cdot ||\phi(s,s')||_{\Lambda^{-1}} \\ &= \underbrace{||\sum_{\tau=1}^N \phi_\tau \epsilon_\tau||_{\Lambda^{-1}}}_{\text{(iii)}} \cdot \sqrt{\phi(s,s')^\top \Lambda^{-1}\phi(s,s')}.\end{aligned} \tag{35}$$

The term (iii) is depend on the randomness of the data collection process of $\mathcal{D}$. To bound this term, we resort to uniform concentration inequalities to upper bound

$$\sup_{V \in \mathcal{V}(R,B,\lambda)} \Big\| \sum_{\tau=1}^N \phi(x_\tau, a_\tau)\cdot \epsilon_\tau(V)\Big\|,$$

where

$$\mathcal{V}(R,B,\lambda) = \{V(s;\nu,\beta,\Sigma) : \mathcal{S} \to [0, V_{\max}] \text{ with} ||\nu|| \le R, \beta \in [0,B], \Sigma \succeq \lambda \cdot I\}, \tag{36}$$

where $V(s;\nu,\beta,\Sigma) = \max_a\{\phi(s,s')^\top \nu - \beta \cdot \sqrt{\phi(s,s')^\top \Sigma^{-1}\phi(s,s')}\}$. For all $\epsilon > 0$, let $\mathcal{N}(\epsilon; R, B, \lambda)$ be the minimal cover if $\mathcal{V}(R,B,\lambda)$. That is, for any function $V \in \mathcal{V}(R,B,\lambda)$, there exists a function $V^\dagger \in \mathcal{N}(\epsilon; R, B, \lambda)$, such that

$$\sup_{s\in\mathcal{S}} |V(s) - V^\dagger(s)| \le \epsilon. \tag{37}$$

Let $R_0 = V_{\max}\sqrt{Nd/\lambda}, B_0 = 2\beta$, it is easy to show that at each iteration, $\widehat{V}^u \in \mathcal{V}(R_0, B_0, \lambda)$. From the definition of $\mathbb{T}$, we have

$$|\mathbb{T}\widehat{V} - \mathbb{T}V^\dagger| = \gamma \left| \int (\widehat{V}(s') - V^\dagger(s'))\, \langle \phi(s,s'), \mu(s')\rangle \, \mathrm{d}s' \right| \le \gamma\epsilon. \tag{38}$$

Then we have

$$|(r + \gamma V - \mathbb{T}V) - (r + \gamma V^\dagger - \mathbb{T}V^\dagger)| \le 2\gamma\epsilon. \tag{39}$$

Let $\epsilon_\tau^\dagger = r(s_\tau, a_\tau) + \gamma V^\dagger(s_{\tau+1}) - \mathbb{T}V^\dagger(s,s')$, we have

$$\begin{aligned}\text{(iii)}^2 = ||\sum_{\tau=1}^N \phi_\tau \epsilon_\tau||_{\Lambda^{-1}}^2 &\le 2||\sum_{\tau=1}^N \phi_\tau \epsilon_\tau^\dagger||_{\Lambda^{-1}}^2 + 2||\sum_{\tau=1}^N \phi_\tau(\epsilon_\tau^\dagger - \epsilon_\tau)||_{\Lambda^{-1}}^2 \\ &\le 2||\sum_{\tau=1}^N \phi_\tau \epsilon_\tau^\dagger||_{\Lambda^{-1}}^2 + 8\gamma^2\epsilon^2 \sum_{\tau=1}^N |\phi_\tau \Lambda^{-1}\phi_\tau| \\ &\le 2||\sum_{\tau=1}^N \phi_\tau \epsilon_\tau^\dagger||_{\Lambda^{-1}}^2 + 8\gamma^2\epsilon^2 N^2/\lambda\end{aligned}$$

It remains to bound $||\sum_{\tau=1}^{N}\phi_\tau\epsilon_\tau^\dagger||^2_{\Lambda^{-1}}$. From the assumption for data collection process, it is easy to show that $\mathbb{E}_{\mathcal{D}}[\epsilon_\tau \,|\, \mathcal{F}_{\tau-1}] = 0$, where $F_{\tau-1} = \sigma(\{(s_i,a_i)_{i=1}^{\tau} \cup (r_i, s_{i+1})_{i=1}^{\tau}\})$ is the σ-algebra generated by the variables from the first τ step. Moreover, since $\epsilon_\tau \le 2V_{\max}$, we have ϵ_τ are $2V_{\max}$-sub-Gaussian conditioning on $F_{\tau-1}$. Then we invoke Lemma F.6 with $M_0 = \lambda\cdot I$ and $M_k = \lambda\cdot I + \sum_{\tau=1}^{k}\phi(x_\tau,a_\tau)\,\phi(x_\tau,a_\tau)^\top$. For the fixed function $V\colon \mathcal{S}\to[0,V_{\max}]$, we have

$$\mathbb{P}_{\mathcal{D}}\left(\Big\|\sum_{\tau=1}^{N}\phi(x_\tau,a_\tau)\cdot\epsilon_\tau(V)\Big\|^2_{\Lambda^{-1}} > 8V_{\max}^2\cdot\log\left(\frac{\det(\Lambda)^{1/2}}{\delta\cdot\det(\lambda\cdot I)^{1/2}}\right)\right)\le\delta \tag{40}$$

for all $\delta\in(0,1)$. Note that $\|\phi(s,s')\|\le 1$ for all $(s,s')\in\mathcal{S}\times\mathcal{S}$ by Definition 2.1. We have

$$\|\Lambda\|_{\text{op}} = \Big\|\lambda\cdot I + \sum_{\tau=1}^{N}\phi(s_\tau,a_\tau)\phi(s_\tau,a_\tau)^\top\Big\|_{\text{op}} \le \lambda + \sum_{\tau=1}^{N}\|\phi(s_\tau,a_\tau)\phi(s_\tau,a_\tau)^\top\|_{\text{op}} \le \lambda + N,$$

where $\|\cdot\|_{\text{op}}$ denotes the matrix operator norm. Hence, it holds that $\det(\Lambda)\le(\lambda+N)^d$ and $\det(\lambda\cdot I)=\lambda^d$, which implies

$$\begin{aligned}
&\mathbb{P}_{\mathcal{D}}\left(\Big\|\sum_{\tau=1}^{N}\phi(s_\tau,a_\tau)\cdot\epsilon_\tau(V)\Big\|^2_{\Lambda_{-1}} > 4V_{\max}^2\cdot\big(2\cdot\log(1/\delta)+d\cdot\log(1+N/\lambda)\big)\right)\\
&\quad\le\mathbb{P}_{\mathcal{D}}\left(\Big\|\sum_{\tau=1}^{N}\phi(s_\tau,a_\tau)\cdot\epsilon_\tau(V)\Big\|^2_{\Lambda_{-1}} > 8V_{\max}^2\cdot\log\left(\frac{\det(\Lambda)^{1/2}}{\delta\cdot\det(\lambda\cdot I)^{1/2}}\right)\right)\le\delta.
\end{aligned}$$

Therefore, we conclude the proof of Lemma F.2.

Applying Lemma F.2 and the union bound, we have

$$\mathbb{P}_{\mathcal{D}}\left(\sup_{V\in\mathcal{N}(\varepsilon)}\Big\|\sum_{\tau=1}^{N}\phi(x^\tau,a^\tau)\cdot\epsilon_\tau(V)\Big\|^2_{\Lambda^{-1}} > 4V_{\max}^2\cdot\big(2\cdot\log(1/\delta)+d\cdot\log(1+N/\lambda)\big)\right)\le\delta\cdot|\mathcal{N}(\varepsilon)|. \tag{41}$$

Recall that

$$\widehat{V}\in\mathcal{V}(R_0,B_0,\lambda),\qquad\text{where }\ R_0 = V_{\max}\sqrt{Nd/\lambda},\ B_0=2\beta,\ \lambda=1,\ \beta = c\cdot dV_{\max}\sqrt{\zeta}. \tag{42}$$

Here $c>0$ is an absolute constant, $\xi\in(0,1)$ is the confidence parameter, and $\zeta=\log(2dV_{\max}/\xi)$ is specified in Algorithm 1. Applying Lemma F.5 with $\varepsilon = dV_{\max}/N$, we have

$$\begin{aligned}
\log|\mathcal{N}(\varepsilon)| &\le d\cdot\log(1+4d^{-1/2}N^{3/2}) + d^2\cdot\log(1+32c^2\cdot d^{1/2}N^2\zeta)\\
&\le d\cdot\log(1+4d^{1/2}N^2) + d^2\cdot\log(1+32c^2\cdot d^{1/2}N^2\zeta).
\end{aligned} \tag{43}$$

By setting $\delta=\xi/|\mathcal{N}(\varepsilon)|$, we have that with probability at least $1-\xi$,

$$\begin{aligned}
&\Big\|\sum_{\tau=1}^{N}\phi(s_\tau,a_\tau)\cdot\epsilon_\tau(\widehat{V})\Big\|^2_{\Lambda^{-1}}\\
&\quad\le 8V_{\max}^2\cdot\big(2\cdot\log(V_{\max}/\xi)+4d^2\cdot\log(64c^2\cdot d^{1/2}N^2\zeta)+d\cdot\log(1+N)+4d^2\big)\\
&\quad\le 8V_{\max}^2 d^2\zeta(4+\log(64c^2)).
\end{aligned} \tag{44}$$

Here the last inequality follows from simple algebraic inequalities. We set $c\ge 1$ to be sufficiently large, which ensures that $36+8\cdot\log(64c^2)\le c^2/4$ on the right-hand side of Equation 44. By Equations Equation 35 and Equation 44, it holds that

$$|\text{(ii)}|\le c/2\cdot dV_{\max}\sqrt{\zeta}\cdot\sqrt{\phi(s,s')^\top\Lambda^{-1}\phi(s,s')} = \beta/2\cdot\sqrt{\phi(s,s')^\top\Lambda^{-1}\phi(s,s')} \tag{45}$$

By Equations Equation 13, Equation 33, Equation 34, and Equation 45, for all $(s,s')\in\mathcal{S}\times\mathcal{S}$, it holds that

$$\big|(\mathbb{T}\widehat{V})(s,s') - (\widehat{\mathbb{T}}\widehat{V})(s,s')\big| \le (V_{\max}\sqrt{d}+\beta/2)\cdot\sqrt{\phi(s,s')^\top\Lambda^{-1}\phi(s,s')} \le \Gamma(s,s') \tag{46}$$

with probability at least $1-\xi$. Therefore, we conclude the proof of Lemma F.2. □

Lemma F.3 (Bounded weight of value function)**.** *Let $V_{max} = r_{max}/(1-\gamma)$. For any function $V : \mathcal{S} \to [0, V_{max}]$, we have*

$$||\nu|| \leq V_{max}\sqrt{d}, ||\widehat{\nu}|| \leq V_{max}\sqrt{\frac{Nd}{\lambda}}.$$

Proof. since

$$\nu^\top \phi(s, s') = \langle M, \phi(s, s')\rangle + \gamma \int V(s')\psi(s')^\top M\phi(s, s')\mathrm{d}s',$$

We have

$$\begin{aligned}\nu &= M + \gamma \int V(s')\psi(s')^\top M\mathrm{d}s' \\ &= r_{\max}\sqrt{d} + \gamma V_{\max}\sqrt{d} \\ &= V_{\max}\sqrt{d}.\end{aligned}$$

For $\widehat{\nu}$, we have

$$\begin{aligned}||\widehat{\nu}|| &= ||\Lambda^{-1}\sum_{\tau=1}^{N}\phi_\tau(r_\tau + \gamma V(s_{\tau+1}))|| \\ &\leq \sum_{\tau=1}^{N}||\Lambda^{-1}\phi_\tau(r_\tau + \gamma V(s_{\tau+1}))|| \\ &\leq V_{\max}\sum_{\tau=1}^{N}||\Lambda^{-1}\phi_\tau|| \\ &\leq V_{\max}\sum_{\tau=1}^{N}\sqrt{\phi_\tau^\top \Lambda^{-1/2}\Lambda^{-1}\Lambda^{-1/2}\phi_\tau} \\ &\leq \frac{V_{\max}}{\sqrt{\lambda}}\sum_{\tau=1}^{N}\sqrt{\phi_\tau^\top \Lambda^{-1}\phi_\tau} \\ &\leq V_{\max}\sqrt{\frac{N}{\lambda}}\sqrt{\mathrm{Tr}(\Lambda^{-1}\sum_{\tau=1}^{T}\phi_\tau\phi_\tau^\top)} \\ &\leq V_{\max}\sqrt{\frac{Nd}{\lambda}}.\end{aligned}$$

□

F.2. Proof of Theorem 3.3

Proof. It is easy to show that the problem can be casted as a linear contextual bandit problem with a misspecified reward function. The error of the reward function is bounded as in Theorem 3.2. Then we can apply the result in Abbasi-Yadkori et al. (2011) to obtain the regret bound, and we omit the detailed proof for simplicity. □

F.3. Proof of Corollary 3.4

Proof. Note that without passive data, the online learning regret will scale as

$$Reg(T) = \frac{2\sqrt{d^2\zeta_2}\cdot r_{\max}}{1-\gamma}\sqrt{T}. \tag{47}$$

Setting the regret bound in Theorem 3.3 to be less than the online regret bound above, we have the result immediately. □

F.4. Technical Lemmas

Lemma F.4 (Linear MDP is Linear Representable)**.** *For a linear MDP, the optimal state-state Q-value function is linear with respect to*

$$\phi(s, s') = [\varphi(s); \psi(s', \pi^\star(s'))]. \tag{48}$$

Proof. From definition we know that the optimal state-action value function is linear with respect to $\phi(s, a)$. Suppose $Q^\star(s, a) = \phi(s, a)^\top w^\star$, then we have

$$\begin{aligned} Q^\star(s, s') &= r(s) + \gamma V(s') && (49)\\ &= \langle \theta, \psi(s) \rangle + \gamma \mathbb{E}_{a' \sim \pi^\star} Q^\star(s', a') && (50)\\ &= \langle \theta, \psi(s) \rangle + \gamma \phi(s, a)^\top w^\star && (51)\\ &= \phi(s, s')^\top [\theta; \gamma w^\star]. && (52) \end{aligned}$$

□

Lemma F.4 indicates that there always exists a linear representation $\phi(s, s')$ for the optimal state-state value function $Q^\star(s, s')$. While the existence of linear representation may not hold for any value function $V(s)$, we can always project the state-state value function $Q(s, s')$ to the linear functions of $\phi(s, s')$, which does not increase the distance to the optimal state-state value function.

Lemma F.5 (ε-Covering Number (Jin et al., 2020))**.** *For all $h \in [H]$ and all $\varepsilon > 0$, we have*

$$\log |\mathcal{N}(\varepsilon; R, B, \lambda)| \le d \cdot \log(1 + 4R/\varepsilon) + d^2 \cdot \log\big(1 + 8d^{1/2} B^2/(\varepsilon^2 \lambda)\big).$$

Proof of Lemma F.5. See Lemma D.6 in (Jin et al., 2020) for a detailed proof. □

Lemma F.6 (Concentration of Self-Normalized Processes (Abbasi-Yadkori et al., 2011))**.** *Let $\{\mathcal{F}_t\}_{t=0}^\infty$ be a filtration and $\{\epsilon_t\}_{t=1}^\infty$ be an $\mathbb{R}$-valued stochastic process such that ϵ_t is $\mathcal{F}_t$-measurable for all $t \ge 1$. Moreover, suppose that conditioning on $\mathcal{F}_{t-1}$, ϵ_t is a zero-mean and σ-sub-Gaussian random variable for all $t \ge 1$, that is,*

$$\mathbb{E}[\epsilon_t \,|\, \mathcal{F}_{t-1}] = 0, \qquad \mathbb{E}\big[\exp(\lambda \epsilon_t) \,\big|\, \mathcal{F}_{t-1}\big] \le \exp(\lambda^2 \sigma^2/2), \qquad \forall \lambda \in \mathbb{R}.$$

Meanwhile, let $\{\phi_t\}_{t=1}^\infty$ be an $\mathbb{R}^d$-valued stochastic process such that ϕ_t is $\mathcal{F}_{t-1}$-measurable for all $t \ge 1$. Also, let $M_0 \in \mathbb{R}^{d \times d}$ be a deterministic positive-definite matrix and

$$M_t = M_0 + \sum_{s=1}^t \phi_s \phi_s^\top$$

for all $t \ge 1$. For all $\delta > 0$, it holds that

$$\Big\| \sum_{s=1}^t \phi_s \epsilon_s \Big\|_{M_t^{-1}}^2 \le 2\sigma^2 \cdot \log\Big(\frac{\det(M_t)^{1/2} \cdot \det(M_0)^{-1/2}}{\delta} \Big)$$

for all $t \ge 1$ with probability at least $1 - \delta$.

Proof. See Theorem 1 of (Abbasi-Yadkori et al., 2011) for a detailed proof. □