Understanding and Addressing the Pitfalls of Bisimulation-based Representations in Offline Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

While bisimulation-based approaches hold promise for learning robust state representations for Reinforcement Learning (RL) tasks, their efficacy in offline RL tasks has not been up to par. In some instances, their performance has even significantly underperformed alternative methods. We aim to understand why bisimulation methods succeed in online settings, but falter in offline tasks. Our analysis reveals that missing transitions in the dataset are particularly harmful to the bisimulation principle, leading to ineffective estimation. We also shed light on the critical role of reward scaling in bounding the scale of bisimulation measurements and of the value error they induce. Based on these findings, we propose to apply the expectile operator for representation learning to our offline RL setting, which helps to prevent overfitting to incomplete data. Meanwhile, by introducing an appropriate reward scaling strategy, we avoid the risk of feature collapse in representation space. We implement these recommendations on two state-of-the-art bisimulation-based algorithms, MICo and SimSR, and demonstrate performance gains on two benchmark suites: D4RL and Visual D4RL. We provide our code in Supplementary Material.

1 Introduction

Reinforcement learning (RL) algorithms often require a significant amount of data to achieve optimal performance [37, 43, 21]. In scenarios where collecting data is costly or impractical, Offline RL methods offer an attractive alternative by learning effective policies from previously collected data [26, 29]. However, capturing the complex structure of the environment from limited data remains a challenge for Offline RL. One promising paradigm to alleviate the issue is to decouple representation learning from policy learning. This involves pre-training the state representation on offline data and then learning the policy upon the fixed representations [45, 42, 38, 47]. Though driven by various motivations, previous methods can be mainly categorized into two classes: i) implicitly shaping the agent's representation of the environment via prediction and control of some aspects of the environment through auxiliary tasks , *e.g.*, maximizing the diversity of visited states [31, 9], exploring attentive contrastive learning on sub-trajectories [45], or capturing temporal information about the environment [42]; ii) utilizing *behavioral metrics*, such as bisimulation metrics [10, 12, 4], to capture complex structure in the environment by measuring the similarity of behavior on the representations [46, 6]. The former methods have proven their effectiveness theoretically and empirically in Offline settings [38, 42, 45], while the adaptability of the latter approaches in the context of limited datasets remains unclear. This paper tackles this question.

Bisimulation-based approaches, as their name suggests, utilize the bisimulation metrics update operator to construct an auxiliary loss and learn robust state representations. These representations

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

encapsulate the behavioral similarities between states by considering the difference between their rewards and dynamics. While the learned representations possess several desirable properties, such as smoothness [18], visual invariance [48, 1, 46], and task adaptation [49, 34, 41, 7], bisimulation-based objectives in most approaches are required to be coupled with the policy improvement procedure [48, 5, 46]. In Offline RL, pretraining state representations via bisimulation-based methods is supposed to be cast as a special case of on-policy bisimulation metric learning where the behavior policy is fixed so that good performance should ensue. However, multiple recent studies [45, 20] suggest that bisimulation-based algorithms yield significantly poorer results on Offline tasks compared to a variety of (self-)supervised objectives.

In this work, we highlight problems with using the bisimulation principle as an objective in Offline settings. We aim to provide a theoretical understanding of the performance gap in bisimulation-based approaches between online and offline settings:"*why do bisimulation approaches perform well in Online RL tasks but tend to fail in Offline RL ones?*" By establishing a connection between the Bellman and bisimulation operators, we uncover that missing transitions, which often occur in Offline settings, can cause the bisimulation principle to be compromised. This means that the bisimulation estimator can be ineffective in finite datasets. Moreover, we notice that the scale of the reward impacts the upper bounds of both the bisimulation measurement[1] fixed point and the value error. This scaling term, if not properly handled, can potentially lead to representation collapse.

To alleviate the aforementioned issues, we propose to learn state representations based on the expectile operator. With this asymmetric operator predicting expectiles of the representation distribution, we can achieve a balance between the behavior measurement and the greedy assignment of the measurement over the dataset. This results in a form of regularization over the bisimulation measurement, thus preventing overfitting to the incomplete data, and implicitly avoiding out-of-distribution estimation errors. Besides, by considering the specific properties of different bisimulation measurements, we investigate the representation collapse issue for the ones that are instantiated with bounded distances (*e.g.*, cosine distance) and propose a way to scale rewards that reduces collapse. We integrate these improvements mainly on two bisimulation-based baselines, MICo [6] and SimSR [46], and show the effectiveness of the proposed modifications.

The primary contributions of this work are as follows:

- We investigate the potential harm of directly applying the bisimulation principle in Offline settings, prove that the bisimulation estimator can be ineffective in finite datasets, and emphasize the essential role of reward scaling.
- We propose theoretically motivated modifications on two representative bisimulation-based baselines, including an expectile-based operator and a tailored reward scaling strategy. These proposed changes are designed to address the challenges encountered when applying the bisimulation principle in offline settings.
- We demonstrate the superior performance our approach yields through an empirical study on two benchmark suites, D4RL [14] and Visual D4RL [32].

2 Related Work

State representation learning in Offline RL Pretraining representations has been recently studied in Offline RL settings, where several studies presented its effectiveness [3, 42, 38, 22]. In this paradigm, we learn state representations on pre-collected datasets before value estimation or policy improvement steps are run. The learned representation can then be used for subsequent policy learning, either online or offline. Some typical auxiliary tasks for pretraining state representations include capturing the dynamical [39] and temporal [42] information of the environment, exploring attentive contrastive learning on sub-trajectories [45], or improving policy performance by applying data augmentations techniques to the pixel-based inputs [8, 32].

Bisimulation-based methods The pioneer works by [19, 30] aim to overcome the curse of dimensionality by defining equivalence relations between states to reduce system complexity. However, these approaches are impractical as they usually demand an exact match of transition distributions.

[1] Since some bisimulation-based approaches do not exactly use metrics but instead of pseudometrics, diffuse metrics or else, we will use the term "measurement" in the following.

To address this issue, [11, 13] propose a bisimulation metric to aggregate similar states. This metric quantifies the similarity between two states and serves as a distance measure to allow efficient state aggregation. Unfortunately, it remains computationally expensive as it requires a full enumeration of states. Later, [4] devise an on-policy bisimulation metric for policy evaluation, providing a scalable method for computing state similarity. Building upon this, [48] develop a metric to learn state representations by modeling the latent dynamic transition as Gaussian. [5] further investigate the independent couple sampling strategy to reduce the computational complexity of representation learning, whereas [46] propose to learn state representations built on the cosine distance to alleviate a representation collapse issue. Despite the promising results obtained, one of the major remaining challenges in this paradigm is its dependency on coupling state representation learning with policy training. This is not always suitable for Offline settings, given that obtaining on-policy reward and transition differences is infeasible due to our inability to gather additional agent-environment interactions. To adapt bisimulation-based approaches to Offline settings, one solution is to consider the policy over the dataset as a specific behavior policy, and then apply the bisimulation principle on it to learn state representations in a pretraining stage, thus disentangling policy training from bisimulation-based learning. Notably, although there exist recent studies [45, 39] investigating the potential of bisimulation-based methods to pretrain state representations, it has not yielded satisfactory results yet [45].

3 Preliminaries

3.1 Offline RL

We consider the standard Markov decision process (MDP) framework, in which the environment is given by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, T, r, \gamma)$, with state space $\mathcal{S}$, action space $\mathcal{A}$, transition function T that decides the next state $s' \sim T(\cdot|s, a)$, reward function $r(s, a)$ bounded by $[R_{\text{min}}, R_{\text{max}}]$, and a discount factor $\gamma \in [0, 1)$. The agent in state $s \in \mathcal{S}$ selects an action $a \in \mathcal{A}$ according to its policy, mapping states to a probability distribution over actions: $a \sim \pi(\cdot|s)$. We make use of the state value function $V^\pi(s) = \mathbb{E}_{\mathcal{M},\pi}\left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t, a_t\right) \mid s_0 = s\right]$ to describe the long term discounted reward of policy π starting at state s. In the sequel, we use T_s^a and r_s^a to denote $T(\cdot|s, a)$ and $r(s, a)$, respectively. In Offline RL, we are given a fixed dataset of environment interactions that include N transition samples, *i.e.* $\mathcal{D} = \{s_i, a_i, s_i', r_i\}_{i=1}^N$. We assume that the dataset $\mathcal{D}$ is composed of trajectories generated i.i.d. under the control of a behavior policy π_β, whose state occupancy is denoted by $\mu_\beta(s)$.

3.2 Bisimulation-based Update Operator

The concept of bisimulation is used to establish equivalence relations on states. This is done recursively by considering two states as equivalent if they have the same distribution over state transitions and the same immediate reward [27, 19]. Since bisimulation considers worst-case differences between states, it commonly results in "pessimistic" outcomes. To address this limitation, the π-bisimulation metric was proposed in [4]. This new metric only considers actions induced by a given policy π rather than all actions when measuring the behavior distance between states:

Theorem 1. *[4] Let $\mathbb{M}$ be the set of all measurements on $\mathcal{S}$. Define $\mathcal{F}^\pi : \mathbb{M} \to \mathbb{M}$ by*

$$\mathcal{F}^\pi(g)(s_i, s_j) = |r_{s_i}^\pi - r_{s_j}^\pi| + \gamma \mathcal{W}(g)\left(T_{s_i}^\pi, T_{s_j}^\pi\right) \tag{1}$$

where $s_i, s_j \in \mathcal{S}$, $r_{s_i}^\pi = \sum_{a\in\mathcal{A}} \pi(a|s_i) r_{s_i}^a$, $T_{s_i}^\pi = \sum_{a\in\mathcal{A}} \pi(a|s_i) T_{s_i}^a$, and $\mathcal{W}(g)$ is the Wasserstein distance with cost function g between distributions. Then $\mathcal{F}^\pi$ has a least fixed point $g_\sim^\pi$, and $g_\sim^\pi$ is a π-bisimulation metric.

Although it is feasible to compute the behavior difference measurement $g_\sim^\pi$ by applying the operator $\mathcal{F}^\pi$ iteratively (which is guaranteed to converge to a fixed point since $\mathcal{F}^\pi$ is a contraction), this approach comes at a high computational complexity due to the Wasserstein distance on the right-hand side of the equation. To tackle this issue, MICo [5] proposed using an independent couple sampling strategy instead of optimizing the overall coupling of the distributions $T_{s_i}^\pi$ and $T_{s_j}^\pi$, resulting in a novel measurement to evaluate the difference between states. Additionally, SimSR [46] further explored the potentiality of combining the cosine distance with bisimulation-based measurements to learn state

representations. Both works can be generalized as:

$$\mathcal{F}^\pi G^\pi(s_i, s_j) = |r^\pi_{s_i} - r^\pi_{s_j}| + \gamma \mathbb{E}_{\substack{s'_i \sim T^\pi_{s_i} \\ s'_j \sim T^\pi_{s_j}}} [G^\pi(s'_i, s'_j)], \tag{2}$$

and $\mathcal{F}^\pi$ has a least fixed point $G^\pi_\sim$[2]. The instantiation of G varies in different approaches [5, 46]. For example, in SimSR [46], the cosine distance is used to instantiate G on the embedding space, and the dynamics difference is computed by the cosine distance between the next-state pair (s'_i, s'_j) sampled from a transition model of the environment. A more detailed description can be found in Appendix C.

Lemma 2. *[5] (**Lifted MDP**) The bisimulation-based update operator $\mathcal{F}^\pi$ for $\mathcal{M}$ is the Bellman evaluation operator for a specific lifted MDP.*

Due to this interpretation of the bisimulation-based update operator as the Bellman evaluation operator in a lifted MDP, we can derive certain conclusions about bisimulation by drawing inspiration from policy evaluation methods. In the next section, we will borrow analytical ideas from [16] to prove that the bisimulation-based objective may be ineffective for finite datasets. We summarize all notations in Appendix A and provide all proofs in Appendix D.

4 Ineffective Bisimulation Estimators in Finite Datasets

The high-level idea of bisimulation-based state representation learning is to learn state embeddings such that when states are projected onto the embedding space, their behavioral similarity is maintained. We denote our parameterized state encoder by $\phi : \mathcal{S} \to \mathbb{R}^n$ and a distance $D(\cdot, \cdot)$ in the embedding space $\mathbb{R}^n$ by $G^\pi_\phi(s_i, s_j) \doteq D(\phi(s_i), \phi(s_j))$. For instance, $D(\cdot, \cdot)$ may be the Łukaszyk–Karmowski distance [5] or the cosine distance [46]. To avoid unnecessary confusion, we defer implementation details to Section 5.

When considering bisimulation-based state representations, the goal is to acquire stable state representations under policy π via the measurement $G^\pi_\sim$. The primary focus is usually to minimize a loss over the *bisimulation error*, denoted by Δ^π_ϕ, which measures the distance between the approximation G^π_ϕ and the fixed point $G^\pi_\sim$:

$$\Delta^\pi_\phi(s_i, s_j) := |G^\pi_\phi(s_i, s_j) - G^\pi_\sim(s_i, s_j)|. \tag{3}$$

However, since the fixed point $G^\pi_\sim$ is unobtainable without full knowledge of the underlying MDP, this approximation error is often unknown. Recall that in Lemma 2, we have shown that we can connect a bisimulation-based update operator to a lifted MDP. Taking inspiration from Bellman evaluation for the value function, we define the *bisimulation Bellman residual* ϵ^π_ϕ as:

$$\epsilon^\pi_\phi(s_i, s_j) := |G^\pi_\phi(s_i, s_j) - \mathcal{F}^\pi G^\pi_\phi(s_i, s_j)|. \tag{4}$$

Then, we can connect the bisimulation Bellman residual with the bisimulation error by the following:

Theorem 3. ***(Bisimulation error upper-bound).*** *Let $\mu_\pi(s)$ denote the stationary distribution over states, let $\mu_\pi(\cdot, \cdot)$ denote the joint distribution over synchronized pairs of states (s_i, s_j) sampled independently from $\mu_\pi(\cdot)$. For any state pair $(s_i, s_j) \in \mathcal{S} \times \mathcal{S}$, the bisimulation error $\Delta^\pi_\phi(s_i, s_j)$ can be upper-bounded by a sum of expected bisimulation Bellman residuals ϵ^π_ϕ:*

$$\Delta^\pi_\phi(s_i, s_j) \leq \frac{1}{1-\gamma} \mathbb{E}_{(s'_i, s'_j) \sim \mu_\pi} \left[\epsilon^\pi_\phi(s'_i, s'_j) \right]. \tag{5}$$

Thereafter, the bisimulation Bellman residual is used as a surrogate objective to approximate the fixed point $G^\pi_\sim$ when learning our state representation. Indeed, the minimization of the bisimulation Bellman residual objective over all pairs $(s'_i, s'_j) \sim \mu_\pi$ leads to the minimization of the corresponding bisimulation error. This ensures that if the expected on-policy bisimulation Bellman residual (*i.e.*, $\mathbb{E}_{\mu_\pi}[\epsilon^\pi_\phi]$, and we will use the term "expected bisimulation residual" in following) minimization objective is zero, then the bisimulation error must be zero for the state pairs under the same policy. However, when the dataset is limited, rather than an infinite transition set covering the whole MDP, minimizing the expected bisimulation residual will no longer be sufficient to guarantee a zero bisimulation error.

[2]For readability, we will conflate the notations G^π and $G^\pi(x, y)$, they are the same if not specified

Proposition 4. ***(The expected bisimulation residual is not sufficient over incomplete datasets).*** *If there exists states s'_i and s'_j not contained in dataset $\mathcal{D}$, where the occupancy $\mu_\pi(s'_i|s_i, a_i) > 0$ and $\mu_\pi(s'_j|s_j, a_j) > 0$ for some $(s_i, s_j) \sim \mu_\pi$, then there exists a bisimulation measurement G^π_ϕ and a constant $C > 0$ such that*

- *For all $(\hat{s_i}, \hat{s_j}) \in \mathcal{D}$, the bisimulation Bellman residual $\epsilon^\pi_\phi(\hat{s_i}, \hat{s_j}) = 0$.*
- *There exists $(s_i, s_j) \in \mathcal{D}$, such that the bisimulation error $\Delta^\pi_\phi(s_i, s_j) = C$.*

As an example, if we only have (s_i, a_i, r, s'_i) and (s_j, a_j, r, s'_j) in a dataset, where both rewards equal to zero for state s_i and s_j, and if we choose $G^\pi_\phi(s_i, s_j) = C$, and $G^\pi_\phi(s'_i, s'_j) = \frac{1}{\gamma}C$, then the bisimulation Bellman residual is $\epsilon^\pi_\phi(s_i, s_j) = 0$, while the bisimulation error $\Delta^\pi_\phi = G^\pi_\phi(s_i, s_j) - 0 = C$ is strictly positive. Note that this failure case does not involve modifying the environment in an extremely adversarial manner, it simply occurs when we are required to estimate the representation of states with subsequent states that are missing from the dataset. Since the distance between the missing states can be arbitrarily large as they are out-of-distribution, directly minimizing the Bellman bisimulation error could achieve the minimal Bellman bisimulation error over the dataset, while not necessarily improving the state representation.

In the context of Offline RL, since the dataset is finite, bisimulation-based representation learning ought to be conceptualized as a pretraining process over the behavior policy π_β of the dataset $\mathcal{D}$. However, the failure case above indicates that applying the bisimulation operator $\mathcal{F}^{\pi_\beta}$ and minimizing the associated Bellman bisimulation error does not necessarily ensure the sufficiency of the learned representation for downstream tasks. Ideally, if we had access to the fixed-point measurement $G^{\pi_\beta}_\sim$, then we could directly minimize the error between the approximation G and the fixed-point $G^{\pi_\beta}_\sim$. However, given the static and incomplete nature of the dataset, acquiring the fixed-point $G^{\pi_\beta}_\sim$ explicitly is not feasible. From another perspective, the failure stems from out-of-distribution estimation errors. Assuming we could estimate the bisimulation exclusively with *in-sample learning*, this issue could be intuitively mitigated. As such, we resort to expectile regression as a regularizer, allowing us to circumvent the need for out-of-sample / unseen state pairs.

5 Method

In this section, we describe how we adapt existing bisimulation-based representation approaches to offline RL. We use the expectile-based operator to learn state representations that optimize the behavior measurement over the dataset, while avoiding overfitting to the incomplete data. In addition, we analyze the impact of reward scaling and propose as a consequence to normalize the reward difference in the bisimulation Bellman residual in order to satisfy the specific nature of different instantiations of the bisimulation measurement while keeping a lower value error. The pseudo-code of our method is shown in Algorithms in Appendix B.

5.1 Expectile-based Bisimulation Operator

The efficacy of expectile regression in achieving *in-sample learning* has already been demonstrated in previous research [25, 33]. Consequently, we will first describe our proposed *expectile*-based operator, and subsequently show how expectile regression can effectively address the aforementioned challenge. Specifically, we consider the update operator as follows:

$$\begin{aligned}
\left(\mathcal{F}^{\pi_\beta}_\tau G^{\pi_\beta}_\phi\right)(s_i, s_j) &:= \arg\min_{G^{\pi_\beta}_\phi} \mathbb{E}_{a_i \sim \pi_\beta(\cdot|s_i), a_j \sim \pi_\beta(\cdot|s_j)} \left[\tau[\hat{\epsilon}]^2_+ + (1-\tau)[-\hat{\epsilon}]^2_+\right], \\
\hat{\epsilon} &= \mathbb{E}_{\substack{s'_i \sim T^{\pi_\beta}_{s_i} \\ s'_j \sim T^{\pi_\beta}_{s_j}}} \big[\underbrace{|r(s_i, a_i) - r(s_j, a_j)| + \gamma G^{\pi_\beta}_{\bar{\phi}}(s'_i, s'_j)}_{\text{target } G} - G^{\pi_\beta}_\phi(s_i, s_j) \big],
\end{aligned} \tag{6}$$

where $\hat{\epsilon}$ is the estimated one-step bisimulation Bellman residual, π_β is the behavior policy, $G_{\bar{\phi}}$ is the target encoder, updated using an exponential moving average, and $[\cdot]_+ = \max(\cdot, 0)$. Since the expectile operator in Equation 6 does not have a closed-form solution, in practice, we minimize it through gradient descent steps:

$$G^{\pi_\beta}_\phi(s_i, s_j) \leftarrow G^{\pi_\beta}_\phi(s_i, s_j) - 2\alpha \mathbb{E}_{a_i \sim \pi_\beta(\cdot|s_i), a_j \sim \pi_\beta(\cdot|s_j)} \left[\tau[\hat{\epsilon}]_+ + (1-\tau)[\hat{\epsilon}]_-\right] \tag{7}$$

where α is the step size. The fixed-point of the measurement obtained using this expectile-based operator is denoted as G_τ. Although the utilization of the *expectile* statistics is well established, its application for estimating bisimulation measurement is not particularly intuitive. In the following, we will show how expectile-based operator can be helpful in addressing the aforementioned issue. First, it is worth noting that when $\tau = 1/2$, this operator becomes the bisimulation expectation of the behavior policy, *i.e.*, $\mathbb{E}_{\mu_{\pi_\beta}}[\hat{\epsilon}]$. Next, we shall consider how this operator performs when $\tau \to 1$. We show that under certain assumptions, our method indeed approximates an "optimal" measurement in terms of the given dataset. We first prove a technical lemma stating that the update operator is still a contraction, and then prove a lemma relating different expectiles, finally we derive our main result regarding the "optimality" of our method.

Lemma 5. *For any $\tau \in [0, 1)$, $\mathcal{F}^\pi_\tau$ is a γ_τ-contraction, where $\gamma_\tau = 1-2\alpha(1-\gamma)\min\{\tau, 1-\tau\} < 1$.*

Lemma 6. *For any $\tau, \tau' \in [0, 1)$ with $\tau' \geq \tau$, and for all $s_i, s_j \in \mathcal{S}$ and any α, we have $G_{\tau'} \geq G_\tau$.*

Theorem 7. *In deterministic MDP and fixed finite dataset, we have:*

$$\lim_{\tau \to 1} G_\tau(s_i, s_j) = \max_{\substack{a_i \in \mathcal{A}, a_j \in \mathcal{A} \\ \text{s.t. } \pi_\beta(a_i|s_i)>0, \pi_\beta(a_j|s_j)>0}} G^*_\sim((s_i, a_i), (s_j, a_j)). \tag{8}$$

where $G^*_\sim((s_i, a_i), (s_j, a_j))$ is a fixed-point measurement constrained to the dataset and defined on the state-action space $\mathcal{S} \times \mathcal{A}$ as

$$G^*_\sim((s_i, a_i), (s_j, a_j)) = |r(s_i, a_i) - r(s_j, a_j)| + \gamma \mathbb{E}_{\substack{s'_i \sim T^{\pi_\beta}_{s_i} \\ s'_j \sim T^{\pi_\beta}_{s_j}}} \left[\max_{\substack{a'_i \in \mathcal{A}, a'_j \in \mathcal{A} \\ \text{s.t. } \pi_\beta(a'_i|s'_i)>0, \pi_\beta(a'_j|s'_j)>0}} G^*_\sim((s'_i, a'_i), (s'_j, a'_j)) \right].$$

Intuitively, $G_\sim((s_i, a_i), (s_j, a_j))$ can be interpreted as a state-action value function $Q(\tilde{s}, \tilde{a})$ in a lifted MDP $\widetilde{M}$, and $G_\sim(s_i, s_j)$ as a state value function $V(\tilde{s})$. We defer the detailed explanation to Appendix E.

Theorem 7 illustrates that, as $\tau \to 1$, we are effectively approximating the maximum $G_\sim((s_i, a_i), (s_j, a_j))$ over actions a'_i, a'_j from the dataset. When we set $\tau = 1$, the expectile-based bisimulation operator achieves fully in-sample learning: we only consider state pairs that have corresponding actions in the dataset. For instance, only when we have $(s'_i, a'_i) \in \mathcal{D}$ and $(s'_j, a'_j) \in \mathcal{D}$, can we apply the measurement of $G^*_\sim$. As such, by manipulating τ, we balance a trade-off between minimizing the expected bisimulation residual (for $\tau = 0.5$) and evaluating $G^*_\sim((s_i, a_i), (s_j, a_j))$ solely on the dataset (for $\tau = 1$), thereby sidestepping the failure case outlined in Proposition 4 in an implicit manner.

5.2 Reward Scaling

Most previous works [4, 48, 5, 46] have overlooked the impact of reward scaling in the bisimulation operator. To demonstrate its importance, we investigate a more general form of the bisimulation operator in Equation 2, given as:

$$\mathcal{F}^\pi G(s_i, s_j) = c_r \cdot |r^\pi_{s_i} - r^\pi_{s_j}| + c_k \cdot \mathbb{E}^\pi_{s'_i, s'_j}[G(s'_i, s'_j)]. \tag{9}$$

We then can derive the following:

$$\begin{aligned} G^\pi_\sim(s_i, s_j) = \mathcal{F}^\pi G^\pi_\sim(s_i, s_j) &= c_r \cdot |r^\pi_{s_i} - r^\pi_{s_j}| + c_k \cdot \mathbb{E}^\pi_{s'_i, s'_j}[G^\pi_\sim(s'_i, s'_j)] \\ &\leq c_r \cdot (R_{\max} - R_{\min}) + c_k \cdot \mathbb{E}^\pi_{s'_i, s'_j}[G^\pi_\sim(s'_i, s'_j)] \\ &\leq c_r \cdot (R_{\max} - R_{\min}) + c_k \cdot \max_{s'_i, s'_j} G^\pi_\sim(s'_i, s'_j). \end{aligned} \tag{10}$$

Accordingly, we have $G^\pi_\sim(s_i, s_j) \leq \frac{c_r \cdot (R_{\max} - R_{\min})}{1 - c_k}$. Adopting the conventional settings of $c_r = 1$ and $c_k = \gamma$ as suggested in [5, 46], could possibly result in a relatively large upper bound of $G^\pi_\sim$ between states. This is due to the common practice of setting γ at 0.99. However, when bisimulation operators are instantiated with bounded distances, e.g., cosine distance, such a setting may be unsuitable. Therefore, it becomes important to tighten the upper bound.

Besides, we can also derive the value bound between the ground truth value function and the approximated value function:

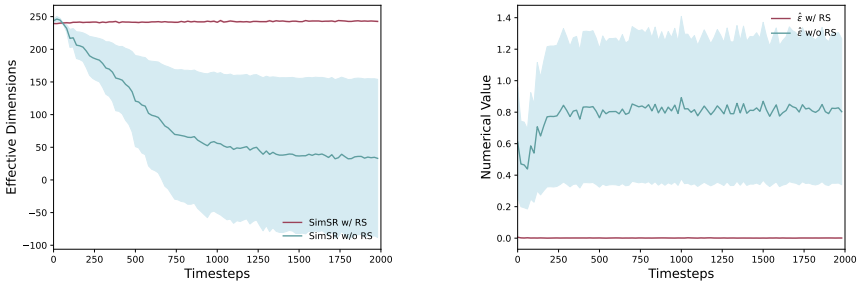

Figure 1: The effectiveness of Reward Scaling (***RS***) in SimSR on halfcheetah-medium-expert-v2, with results averaged on 3 random seeds. (**Left**) Effective Dimension [47] comparison: without ***RS***, there is a significant reduction in the effective dimension, accompanied by a marked increase in instability as training progresses. (**Right**) Numerical value comparison of estimated bisimulation Bellman residual: $\hat{\epsilon}$ is persistently greater than 0 in the absence of ***RS***, which indicates that target G is invariably larger than G_ϕ, suggesting that G_ϕ does not achieve steady convergence.

Theorem 8. *(Value bound based on on-policy bisimulation measurements in terms of approximation error). Given an MDP $\widetilde{\mathcal{M}}$ constructed by aggregating states in an ω-neighborhood, and an encoder ϕ that maps from states in the original MDP $\mathcal{M}$ to these clusters, the value functions for the two MDPs are bounded as*

$$\left|V^{\pi}(s) - \widetilde{V}^{\pi}(\phi(s))\right| \leq \frac{2\omega + \hat{\Delta}}{c_r(1-\gamma)}. \tag{11}$$

where $\hat{\Delta} := \|\hat{G}^{\pi}_{\sim} - \hat{G}^{\pi}_{\phi}\|_{\infty}$ is the approximation error.

In essence, Equation 10 and Theorem 8 reveal that: (i) there is a positive correlation between the reward scale c_r and the upper bound of the fixed-point $G^{\pi}_{\sim}$, and (ii) a larger reward scale c_r facilitates a more accurate approximation of the value function $\widetilde{V}^{\pi}(\phi(s))$ to its ground-truth value $V^{\pi}(s)$. It is important to note that c_r also impacts the value of $\hat{\Delta}$, as depicted in Figure 1(Right)[3]. Therefore, it is crucial to first ensure the alignment with the instantiation of the bisimulation measurement, and then choose the largest possible c_r to minimize the value error. For instance, as the SimSR operator [46] uses the cosine distance, $c_k = \gamma$ is predetermined. We should thus set $c_r \in [0, 1-\gamma]$, and apply min-max normalization to the reward function. This can make $G^{\pi}_{\sim} \leq 1$ and therefore be consistent with the maximum value of 1 of the cosine distance. To achieve a tighter bound in Equation11, we should then maximize the reward scale, setting c_r to $1-\gamma$. Figure 1 illustrates the effectiveness of this reward scaling.

6 Experiments

6.1 Performance Comparison in D4RL Benchmark

Implementation Details We analyze our proposed method on the D4RL benchmark [14] of OpenAI gym MuJoCo tasks [44] which includes a variety of datasets that have been commonly used in the Offline RL community. To illustrate the effectiveness of our method, we implement it on top of two bisimulation-based approaches, **MICo** [5] and **SimSR** [46]. It is worth noting that there are two versions of SimSR depending on its use of a latent dynamics model: SimSR_basic follows the dynamics that the environment provides, and SimSR_full constructs latent dynamics for sampling successive latent states. We opt for SimSR_basic as our backbone, as it exhibits superior and more stable performance in the D4RL benchmark tasks compared to SimSR_full. Additionally, to explore the impact of bisimulation-based representation learning on the downstream performance of policy learning, we build these approaches on top of the Offline RL method **TD3BC** [15]. We examine three

[3]Despite Figure 1(Right) depicting the approximate residual $\hat{\epsilon}$, we have drawn a connection between ϵ^{π}_{ϕ} and Δ^{π}_{ϕ} in the Appendix, which can reflect the possible situations for $\hat{\Delta}$.

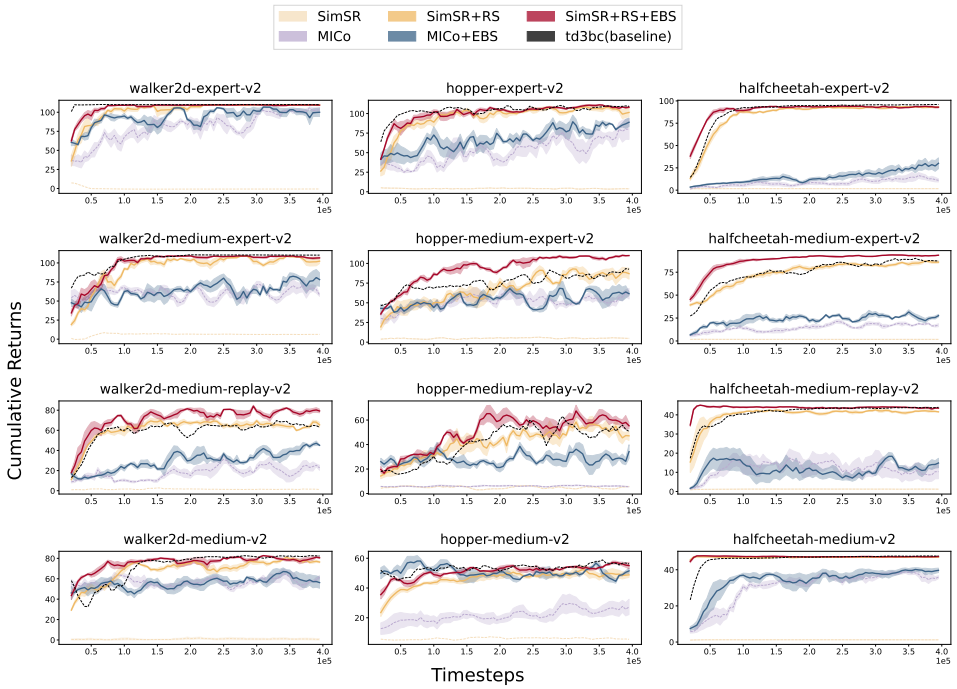

Figure 2: Performance comparison on 12 D4RL tasks over 10 seeds with one standard error shaded in the default setting. For every seed, the average return is computed every 10,000 training steps, averaging over 10 episodes. The horizontal axis indicates the number of transitions trained on. The vertical axis indicates the normalized average return.

environments: halfcheetah, hopper, and walker2d, with four datasets per task: expert, medium-expert, medium-replay, and medium. We first pretrain the encoder during $100k$ timesteps, then freeze it, pass the raw state through the frozen encoder to obtain the representations that serve as input for the Offline RL algorithm. Further details on the experiment setup are included in Appendix F.

Analysis Figure 2 illustrates the performance of two approaches and their variants in the D4RL tasks. We use ***EBS*** to represent the scheme of employing the expectile-based operator, while ***RS*** denotes the reward scaling scheme. The latter includes both min-max reward normalization and penalization coefficient with $(1-\gamma)$ in the bisimulation operator. As discussed in Section 5.2, the role of reward scaling varies depending on the specific instantiation of G[4]. We observe that without ***RS***, SimSR almost fails in every dataset, which aligns with our understanding of the critical role reward scaling plays. The results also illustrate that ***EBS*** effectively enhances the downstream performance of the policy for both SimSR and MICo. It is noteworthy that in this experiment, we set $\tau = 0.6$ for the expectile in SimSR and $\tau = 0.7$ in MICo across all datasets, demonstrating the robustness of this hyperparameter. Regarding SimSR, when ***RS*** is applied (***SimSR+RS***), the performance is comparable to the TD3BC baseline, while the incorporation of the expectile-based operator (***SimSR+RS+EBS***) further enhances final performance

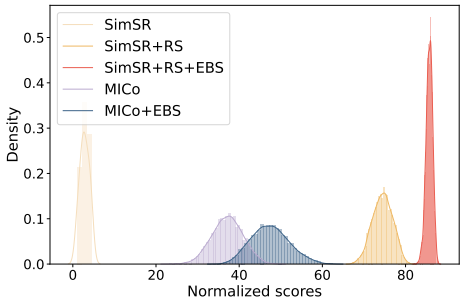

Figure 3: Bootstrapping distributions for uncertainty in IQM (*i.e.*, inter-quartile mean) measurement, following from the performance criterion in [2].

[4]Since MICo does not necessitate a particular upper bound, RS may be harmful to its performance. Our experiments have substantiated this observation, leading us to exclude the MICo+RS results from Figure 2.

Table 1: Performance comparison with several other baselines on V-D4RL benchmark, averaged on 3 random seeds.

Dataset	CURL	DRIMLC	HOMER	ICM	MICo → MICo+EBS	SimSR → SimSR+RS+EBS
cheetah-run-medium	392	**524**	475	365	177 → 449 (↗ 272)	391 → 491(↗ 100)
walker-walk-medium	452	425	439	358	450 → 447 (—)	443 → **480**(↗ 37)
cheetah-run-medium-replay	271	395	306	251	335 →357 (↗ 22)	374 → **462**(↗ 88)
walker-walk-medium-replay	265	235	**283**	167	207 → 240 (↗ 33)	197 → 240(↗ 43)
cheetah-run-medium-expert	348	403	383	280	282 → 341 (↗ 59)	360 → **547**(↗ 187)
walker-walk-medium-expert	729	399	781	606	586 → 635(↗ 49)	755 → **845**(↗ 90)
cheetah-run-expert	200	310	218	237	308 → 331(↗ 23)	409 → **454**(↗ 45)
walker-walk-expert	769	427	686	**850**	370 → 447 (↗ 77)	578 → 580 (—)
total	3426	3118	3571	3114	2715 → 3253 (↗ 538)	3507 → **4043** (↗ 536)

and sample efficiency. Besides, we additionally present the IQM normalized return of all variants in Figure 3, illustrating our performance gains over the backbones. Further, we have also constructed an ablation study to investigate the impact of different settings of τ, the results show that a suitable expectile τ is crucial for control tasks. We present the corresponding results in Appendix E.

6.2 Performance Comparison in V-D4RL Benchmark

Implementation details We also evaluate our method on a visual observation setting of DMControl suite (DMC) tasks, V-D4RL benchmark [32]. Similar to the previous experiment, we add the proposed schemes on top of MICo and SimSR. In the experiments, we notice that the latent dynamics modeling can help to boost performance for the visual setting, hence we use SimSR_full as the backbone. Additionally, we also notice that MICo often gives really poor performance in the V-D4RL benchmark, while adding latent dynamics alleviates the issue. Therefore, we boost MICo with explicit dynamics modeling for a fair comparison. To compare the performance with the other representation approaches, we include 4 competitive representation learning approaches for Offline RL, including DRIML [35], HOMER [36], CURL [28], and Inverse model [40]. Detailed descriptions of these approaches can be found in Appendix G.

Analysis We evaluate all aforementioned approaches by integrating the pre-trained encoder from each into an Offline RL method DrQ+BC [32], which combines data augmentation techniques with TD3BC. The results in Table 1 illustrate the effectiveness of our proposed method, the numerical improvements are underlined with red upward arrows. Compared to the other baselines, while ***SimSR+RS+EBS*** does not achieve the highest score in all datasets, it achieves the best overall performance. Besides, our modifications on MICo and SimSR consistently show significant improvements. This indicates that our proposed method is not only applicable to raw-state inputs but also compatible with pixel-based observations.

7 Discussion

Limitations and Future Work While τ remains constant in our D4RL experiments, optimal performance may arise under different τ settings, contingent on the specific attributes of the dataset. Therefore, to yield the best outcomes, one might need to set various τ to identify the most suitable value. However, this process could consume substantial computational resources. Another area of potential study involves evaluating the effectiveness of our approach in off-policy settings, given that off-policy settings may also lead to similar failure cases.

Conclusion In this work, we highlight the effectiveness of the bisimulation operator over incomplete datasets and emphasize the crucial role of reward scaling in Offline settings. By employing the expectile operator in bisimulation, we manage to strike a balance between behavior measurement and greedy assignment of the measurement over datasets. We also propose a reward scaling strategy to reduce the risk of representation collapse in specific bisimulation-based measurements. Empirical studies show the effectiveness of our proposed modifications.

References

[1] Rishabh Agarwal, Marlos C. Machado, Pablo Samuel Castro, and Marc G. Bellemare. Contrastive behavioral similarity embeddings for generalization in reinforcement learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[2] Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 29304–29320, 2021.

[3] Sanjeev Arora, Simon S. Du, Sham M. Kakade, Yuping Luo, and Nikunj Saunshi. Provable representation learning for imitation learning via bi-level optimization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 367–376. PMLR, 2020.

[4] Pablo Samuel Castro. Scalable methods for computing state similarity in deterministic markov decision processes. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 10069–10076. AAAI Press, 2020.

[5] Pablo Samuel Castro, Tyler Kastner, Prakash Panangaden, and Mark Rowland. Mico: Improved representations via sampling-based state similarity for markov decision processes. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 30113–30126, 2021.

[6] Pablo Samuel Castro, Tyler Kastner, Prakash Panangaden, and Mark Rowland. Mico: Learning improved representations via sampling-based state similarity for markov decision processes. *CoRR*, abs/2106.08229, 2021.

[7] Pablo Samuel Castro and Doina Precup. Using bisimulation for policy transfer in mdps. In Maria Fox and David Poole, editors, *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*. AAAI Press, 2010.

[8] Xin Chen, Sam Toyer, Cody Wild, Scott Emmons, Ian Fischer, Kuang-Huei Lee, Neel Alex, Steven H Wang, Ping Luo, Stuart Russell, Pieter Abbeel, and Rohin Shah. An empirical investigation of representation learning for imitation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[9] Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[10] Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. In Deborah L. McGuinness and George Ferguson, editors, *Proceedings of the Nineteenth National Conference on Artificial Intelligence, Sixteenth Conference on Innovative Applications of Artificial Intelligence, July 25-29, 2004, San Jose, California, USA*, pages 950–951. AAAI Press / The MIT Press, 2004.

[11] Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. In David Maxwell Chickering and Joseph Y. Halpern, editors, *UAI '04, Proceedings of the 20th Conference in Uncertainty in Artificial Intelligence, Banff, Canada, July 7-11, 2004*, pages 162–169. AUAI Press, 2004.

[12] Norm Ferns, Prakash Panangaden, and Doina Precup. Bisimulation metrics for continuous markov decision processes. *SIAM J. Comput.*, 40(6):1662–1714, 2011.

[13] Norman Ferns and Doina Precup. Bisimulation metrics are optimal value functions. In Nevin L. Zhang and Jin Tian, editors, *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence, UAI 2014, Quebec City, Quebec, Canada, July 23-27, 2014*, pages 210–219. AUAI Press, 2014.

[14] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4RL: datasets for deep data-driven reinforcement learning. *CoRR*, abs/2004.07219, 2020.

[15] Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 20132–20145, 2021.

[16] Scott Fujimoto, David Meger, Doina Precup, Ofir Nachum, and Shixiang Shane Gu. Why should I trust you, bellman? the bellman error is a poor replacement for value error. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 6918–6943. PMLR, 2022.

[17] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1582–1591. PMLR, 2018.

[18] Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G. Bellemare. Deepmdp: Learning continuous latent space models for representation learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 2170–2179. PMLR, 2019.

[19] Robert Givan, Thomas L. Dean, and Matthew Greig. Equivalence notions and model minimization in markov decision processes. *Artif. Intell.*, 147(1-2):163–223, 2003.

[20] Pengjie Gu, Mengchen Zhao, Chen Chen, Dong Li, Jianye Hao, and Bo An. Learning pseudometric-based action representations for offline reinforcement learning. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvári, Gang Niu, and Sivan Sabato, editors, *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, pages 7902–7918. PMLR, 2022.

[21] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1856–1865. PMLR, 2018.

[22] Riashat Islam, Manan Tomar, Alex Lamb, Yonathan Efroni, Hongyu Zang, Aniket Didolkar, Dipendra Misra, Xin Li, Harm van Seijen, Remi Tachet des Combes, and John Langford. Agent-controller representations: Principled offline rl with rich exogenous information, 2022.

[23] Sham M. Kakade and John Langford. Approximately optimal approximate reinforcement learning. In Claude Sammut and Achim G. Hoffmann, editors, *Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002), University of New South Wales, Sydney, Australia, July 8-12, 2002*, pages 267–274. Morgan Kaufmann, 2002.

[24] Mete Kemertas and Tristan Aumentado-Armstrong. Towards robust bisimulation metric learning. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 4764–4777, 2021.

[25] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[26] Sascha Lange, Thomas Gabel, and Martin A. Riedmiller. Batch reinforcement learning. In Marco A. Wiering and Martijn van Otterlo, editors, *Reinforcement Learning*, volume 12 of *Adaptation, Learning, and Optimization*, pages 45–73. Springer, 2012.

[27] Kim Guldstrand Larsen and Arne Skou. Bisimulation through probabilistic testing. In *Conference Record of the Sixteenth Annual ACM Symposium on Principles of Programming Languages, Austin, Texas, USA, January 11-13, 1989*, pages 344–352. ACM Press, 1989.

[28] Michael Laskin, Aravind Srinivas, and Pieter Abbeel. CURL: contrastive unsupervised representations for reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 5639–5650. PMLR, 2020.

[29] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *CoRR*, abs/2005.01643, 2020.

[30] Lihong Li, Thomas J. Walsh, and Michael L. Littman. Towards a unified theory of state abstraction for mdps. In *International Symposium on Artificial Intelligence and Mathematics, ISAIM 2006, Fort Lauderdale, Florida, USA, January 4-6, 2006*, 2006.

[31] Hao Liu and Pieter Abbeel. Behavior from the void: Unsupervised active pre-training. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 18459–18473, 2021.

[32] Cong Lu, Philip J. Ball, Tim G. J. Rudner, Jack Parker-Holder, Michael A. Osborne, and Yee Whye Teh. Challenges and opportunities in offline reinforcement learning from visual observations. *CoRR*, abs/2206.04779, 2022.

[33] Xiaoteng Ma, Yiqin Yang, Hao Hu, Jun Yang, Chongjie Zhang, Qianchuan Zhao, Bin Liang, and Qihan Liu. Offline reinforcement learning with value-based episodic memory. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[34] Bogdan Mazoure, Ahmed M. Ahmed, R. Devon Hjelm, Andrey Kolobov, and Patrick MacAlpine. Cross-trajectory representation learning for zero-shot generalization in RL. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

[35] Bogdan Mazoure, Remi Tachet des Combes, Thang Doan, Philip Bachman, and R. Devon Hjelm. Deep reinforcement and infomax learning. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[36] Dipendra Misra, Mikael Henaff, Akshay Krishnamurthy, and John Langford. Kinematic state abstraction and provably efficient rich-observation reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6961–6971. PMLR, 2020.

[37] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin A. Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nat.*, 518(7540):529–533, 2015.

[38] Ofir Nachum and Mengjiao Yang. Provable representation learning for imitation with contrastive fourier features. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 30100–30112, 2021.

[39] Ofir Nachum and Mengjiao Yang. Provable representation learning for imitation with contrastive fourier features. *CoRR*, abs/2105.12272, 2021.

[40] Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 2778–2787. PMLR, 2017.

[41] Anirban Santara, Rishabh Madan, Pabitra Mitra, and Balaraman Ravindran. Extra: Transfer-guided exploration. In Amal El Fallah Seghrouchni, Gita Sukthankar, Bo An, and Neil Yorke-Smith, editors, *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems, AAMAS '20, Auckland, New Zealand, May 9-13, 2020*, pages 1987–1989. International Foundation for Autonomous Agents and Multiagent Systems, 2020.

[42] Max Schwarzer, Nitarshan Rajkumar, Michael Noukhovitch, Ankesh Anand, Laurent Charlin, R. Devon Hjelm, Philip Bachman, and Aaron C. Courville. Pretraining representations for data-efficient reinforcement learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 12686–12699, 2021.

[43] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy P. Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nat.*, 550(7676):354–359, 2017.

[44] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2012*, pages 5026–5033. IEEE, 2012.

[45] Mengjiao Yang and Ofir Nachum. Representation matters: Offline pretraining for sequential decision making. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 11784–11794. PMLR, 2021.

[46] Hongyu Zang, Xin Li, and Mingzhong Wang. Simsr: Simple distance-based state representations for deep reinforcement learning. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 8997–9005. AAAI Press, 2022.

[47] Hongyu Zang, Xin Li, Jie Yu, Chen Liu, Riashat Islam, Remi Tachet des Combes, and Romain Laroche. Behavior prior representation learning for offline reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023.

[48] Amy Zhang, Rowan Thomas McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[49] Amy Zhang, Shagun Sodhani, Khimya Khetarpal, and Joelle Pineau. Learning robust state abstractions for hidden-parameter block mdps. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

Appendix

Contents

A Notation

Table 2 summarizes our notation.

Table 2: Table of Notation.

Notation	Meaning	Notation	Meaning
$\mathcal{M}$	MDP	$\widetilde{\mathcal{M}}$	Lifted MDP (auxiliary MDP)
$\mathcal{S}$	state space	$\mathcal{A}$	action space
T	transition function	r	reward function
γ	discount factor	π	policy of the agent
$V^\pi(s)$	state value function given policy π	$\mathcal{D}$	dataset
π_β	behavior policy	$\mu_\beta(s)$	state occupancy of the dataset
$\mathcal{F}^\pi$	on-policy bisimulation operator	$g^\pi_\sim$	π-bisimulation metric
$D(\cdot,\cdot)$	a specific distance	$G^\pi_\sim$	fixed point of MICo and SimSR
ϕ	state encoder	$G^\pi_\phi(s_i,s_j)$	parameterized bisimulation measurement
Δ^π_ϕ	bisimulation error	ϵ^π_ϕ	bisimulation Bellman residual
$\mu_\pi(s)$	stationary distribution over states on policy π	μ_π	the distribution over pairs of states
$\mathbb{E}_{\mu_\pi}[\epsilon^\pi_\phi]$	expected on-policy bisimulation Bellman residual	$\mathcal{F}^{\pi_\beta}$	behavior bisimulation operator
τ	expectile term	γ_τ	discount factor with expectile
$\mathcal{F}^{\pi_\beta}_\tau$	behavior bisimulation operator with expectile	$\hat{\epsilon}$	estimated one-step residual
$G_{\bar{\phi}}$	bisimulation measurement parameterized by target encoder	$G_\sim(s_i,a_i,s_j,a_j)$	a measurement on state-action space
$G^*_\sim(s_i,a_i,s_j,a_j)$	maximum measurement constrained to dataset	c_r	scale term of reward in bisimulation
c_k	scale term of transition in bisimulation	$\widetilde{V}^\pi(\phi(s))$	value function based on state encoder
ω	distance bound of aggregating neighbor	$\hat{\Delta}$	approximation error of bisimulation measurement

B Algorithm

We provide the algorithm in Algorithm 1, and a pytorch-like implementation build on top of SimSR in Algorithm 2.

Algorithm 1 Proposed Implementation

Stage 1 Preprocessing:
Min-Max reward normalization: $\bar{r} = \frac{r - r_{\min}}{r_{\max} - r_{\min}}$
Stage 2 Pretraining the encoder:
Initialize encoder parameter ϕ, expectile τ, learning rate α, discount factor γ.
for each gradient step **do**
  Apply reward scaling when computing $\hat{\epsilon}$:

$$\hat{\epsilon} = (1-\gamma)|\bar{r}(s_i,a_i) - \bar{r}(s_j,a_j)| + \gamma G^{\pi_\beta}_{\bar{\phi}}(s'_i,s'_j) - G^{\pi_\beta}_{\phi}(s_i,s_j) \tag{12}$$

  Update encoder ϕ:

$$\phi \leftarrow \phi - 2\alpha \mathbb{E}_{a_i\sim\pi_\beta(\cdot|s_i),a_j\sim\pi_\beta(\cdot|s_j)}\left[\tau[\hat{\epsilon}]_+ + (1-\tau)[\hat{\epsilon}]_-\right] \tag{13}$$

end for
Stage 3 Training value function and policy network:
Initialize value function parameter ψ, policy network parameter θ, learning rate λ_V and λ_π.
for each gradient step **do**
  Sample tuple $(s,a,s',\bar{r})$ from dataset $\mathcal{D}$
  Encode the states to representation space: $z = \phi(s), z' = \phi(s')$
  Update value function with $(z,a,z',\bar{r})$:

$$\psi = \psi - \lambda_V \nabla_\psi \mathcal{L}_V(\psi). \tag{14}$$

  Update policy network with $(z,a,z',\bar{r})$:

$$\theta = \theta - \lambda_\pi \nabla_\theta \mathcal{L}_\pi(\theta). \tag{15}$$

end for

Algorithm 2 SimSR+RS+EBS Pseudocode, PyTorch-like

```
class ReplayBuffer(object):
    def __init__(self):
        ...
        self.reward_normalization()
        ...
    ...
    def reward_normalization(self):
        r_max = self.reward.max()
        r_min = self.reward.min()
        self.reward = (self.reward - r_min) / (r_max - r_min)

def compute_distance(features_a, features_b):
    similarity_matrix = torch.matmul(features_a, features_b.T)
    dis = 1-similarity_matrix
    return dis

def expectile_loss(diff, expectile):
    weight = torch.where(diff > 0, expectile, (1 - expectile))
    return weight * (diff ** 2)

# encoder: mlp, encoder network, the output is l2-normalized
# target_encoder: mlp, same as encoder, updated by EMA
# discount: discount factor $\gamma$
# slope: expectile $\tau$
def compute_ebs_loss(encoder, target_encoder, replay_buffer, batch_size, discount, slope):
    observation, action, reward, discount, next_observation = replay_buffer.sample(batch_size) # sample a
         batch of tuples from replay buffer
    latent_state = encoder(observation)
    latent_next_state = target_encoder(next_observation)
    r_diff = (1 - discount) * torch.abs(reward.T - reward)
    next_diff = compute_distance(latent_next_state, latent_next_state)
    z_diff = compute_distance(latent_state, latent_state)
    bisimilarity = r_diff + discount * next_diff

    encoder_loss = expectile_loss(bisimilarity.detach() - z_diff, slope)
    encoder_loss = encoder_loss.mean()
    return encoder_loss
```

C Technical backgrounds

C.1 Bisimulation metric

Bisimulation measures equivalence relations on MDPs with a recursive form: two states are deemed equivalent if they share the equivalent distributions over the next equivalent states and they have the same immediate reward [27, 19]. However, since bisimulation considers equivalence for all actions, including bad ones, it commonly results in "pessimistic" outcomes. Instead, [4] developed π-bisimulation which removes the requirement of considering each action and only needs to consider the actions induced by a policy π.

Definition 9. *[4] Given an MDP $\mathcal{M}$, an equivalence relation $E^\pi \subseteq \mathcal{S} \times \mathcal{S}$ is a π-bisimulation relation if whenever $(\mathbf{s}, \mathbf{u}) \in E^\pi$ the following properties hold:*

1. $r(s, \pi) = r(u, \pi)$
2. $\forall C \in \mathcal{S}_{E^\pi}, T(C|s, \pi) = T(C|u, \pi)$

where $\mathcal{S}_{E^\pi}$ is the state space $\mathcal{S}$ partitioned into equivalence classes defined by E^π. Two states $s, u \in S$ are π-bisimilar if there exists a π-bisimulation relation E^π such that $(s, u) \in E^\pi$.

However, π-bisimulation is still too stringent to be applied at scale as π-bisimulation relation emphasizes the equivalence is a binary property: either two states are equivalent or not, thus becoming too sensitive to perturbations in the numerical values of the model parameters. The problem becomes even more prominent when deep frameworks are applied.

Thereafter, they proposed a π-bisimulation metric to leverage the absolute value between the immediate rewards w.r.t. two states and the 1-Wasserstein distance ($\mathcal{W}_1$) between the transition distributions conditioned on the two states and the policy π to formulate such measurement:

Theorem 10. *Define $\mathcal{F}^\pi : \mathbb{M} \to \mathbb{M}$ by $\mathcal{F}^\pi(d)(s, u) = |R^\pi_s - R^\pi_u| + \gamma \mathcal{W}_1(d)(T^\pi_s, T^\pi_u)$, then $\mathcal{F}^\pi$ has a least fixed point $d^\pi_\sim$, and $d^\pi_\sim$ is a π-bisimulation metric.*

Although the Wasserstein distance is a powerful metric to calculate the distance between two probability distributions, it requires to enumerate all states which is impossible in RL tasks of continuous state space. Various extensions have been proposed [48, 5, 46] to reduce the computational complexity. DBC [48] extend bisimulation metrics to learn state representation, via minimizing the ℓ_1-norm distance of representations and the bisimulation metrics, meanwhile modeling the latent dynamics as Gaussian and utilizing W_2 distance to compute it, which can be formulated as a closed-form result. However, DBC has several issues like loss function mismatch and specific requirements for Gaussian modeling, which limits its application and performance.

C.2 MICo distance

MICo distance [5], tackles the above issue by restricting the coupling class to the independent coupling to avoid intractable Wasserstein distance computation. The MICo operator and its associated theoretical guarantee are given as:

Theorem 11. *[5] Given a policy π, MICo distance $\mathcal{F}^\pi$ is defined as:*

$$\mathcal{F}^\pi U(s_i, s_j) = |r^\pi_{s_i} - r^\pi_{s_j}| + \gamma \mathbb{E}_{s'_i \sim T^\pi_{s_i}, s'_j \sim T^\pi_{s_j}} [U(s'_i, s'_j)] \tag{16}$$

has a fixed point U^π.

By considering the Wasserstein distance in the definition of bisimulation metrics can be upper-bounded by taking a restricted class of couplings of the transition distributions, MICo restricts the coupling class precisely to the singleton containing the independent coupling, utilizing the Independent Couple sampling strategy to bypass the computation of the Wasserstein distance. However, MICo distance U requires to be a Łukaszyk-Karmowski metric, which does not satisfy the identity of indiscernibles. As a result, the approximated distance on the learned embedding space based on the MICo distance, which involves a Łukaszyk-Karmowski metric to measure the distance between dynamics, may suffer from the violation issue of the identity of indiscernibles.

C.3 SimSR operator

To avoid the potential representation collapse, SimSR [46] develop a more concise update operator to learn state representation more effectively. Coupling with cosine distance, SimSR defines its operator as:

Theorem 12. *[46] Given a policy π, Simple State Representation (SimSR) is updated as:*

$$\mathcal{F}^\pi \overline{cos}_\phi(s_i, s_j) = |r^\pi_{s_i} - r^\pi_{s_j}| + \gamma \mathbb{E}_{s'_i \sim T^\pi_{s_i}, s'_j \sim T^\pi_{s_j}} [\overline{cos}_\phi(s'_i, s'_j)] \tag{17}$$

has the same fixed point as MICo.

Further, considering the latent dynamics can be beneficial to representation learning, they additionally develop a form of operator including dynamics modeling:

Theorem 13. *[46] Given a policy π, and a latent dynamics model $\hat{T}$, SimSR is updated as*

$$\mathcal{F}^\pi \overline{cos}_\phi(s_i, s_j) = |r^\pi_{s_i} - r^\pi_{s_j}| + \gamma \mathbb{E}_{z'_i \sim \hat{T}^\pi_{\phi(s_i)}, z'_j \sim \hat{T}^\pi_{\phi(s_j)}} [\overline{cos}(z'_i, z'_j)]. \tag{18}$$

If latent dynamics are specified, $\mathcal{F}^\pi$ has a fixed point.

When considering MICo distance and the basic version of SimSR, we can notice that they have a similar recursive iteration formulation. And therefore both works can be generalized under:

$$\mathcal{F}^\pi G^\pi(s_i, s_j) = |r^\pi_{s_i} - r^\pi_{s_j}| + \gamma \mathbb{E}_{\substack{s'_i \sim T^\pi_{s_i} \\ s'_j \sim T^\pi_{s_j}}} [G^\pi(s'_i, s'_j)], \tag{19}$$

while the instantiation of G varies in these two approaches.

C.4 Lifted MDP

The connection between bisimulation-based operators and lifted MDP can be referred to [5]. We provide the corresponding Lemma here for reference.

Lemma 2. ***(Lifted MDP)*** *The bisimulation-based update operator $\mathcal{F}^\pi$ for $\mathcal{M}$, is the Bellman evaluation operator for a specific lifted MDP.*

Proof. Given the MDP specified by the tuple $(\mathcal{S}, \mathcal{A}, T, R)$, we construct a lifted MDP $(\widetilde{\mathcal{S}}, \widetilde{\mathcal{A}}, \widetilde{T}, \widetilde{R})$, by taking the state space to be $\widetilde{\mathcal{S}} = \mathcal{S}^2$, the action space to be $\widetilde{\mathcal{A}} = \mathcal{A}^2$, the transition dynamics to be given by $\widetilde{T}^{\tilde{a}}_{\tilde{s}}(\tilde{s}') = \widetilde{T}^{(a_i,a_j)}_{(s_i,s_j)}((s'_i, s'_j)) = T^{a_i}_{s_i}(s'_i)T^{a_j}_{s_j}(s'_j)$ for all $(s_i, s_j), (s'_i, s'_j) \in \mathcal{S}^2$, $a_i, a_j \in \mathcal{A}$, and the action-independent rewards to be $\widetilde{R}_{\tilde{s}} = \widetilde{R}_{(s_i,s_j)} = |r^\pi_{s_i} - r^\pi_{s_j}|$ for all $s_i, s_j \in \mathcal{S}$. The Bellman evaluation operator $\widetilde{\mathcal{F}}^{\tilde{\pi}}$ for this lifted MDP at discount rate γ under the policy $\tilde{\pi}(\tilde{a}|\tilde{s}) = \tilde{\pi}(a_i, a_j|s_i, s_j) = \pi(a_i|s_i)\pi(a_j|s_j)$ is given by (for all $G^\pi \in \mathbb{R}^{\mathcal{S}\times\mathcal{S}}$ and $(s_i, s_j) \in \mathcal{S} \times \mathcal{S}$):

$$
\begin{aligned}
(\widetilde{\mathcal{F}}^{\tilde{\pi}}\tilde{G}^\pi)(\tilde{s}) &= \widetilde{R}_{\tilde{s}} + \gamma \sum_{\tilde{s}' \in \widetilde{\mathcal{S}}} \widetilde{T}^{\tilde{a}}_{\tilde{s}}(\tilde{s}')\tilde{\pi}(\tilde{a}|\tilde{s})\tilde{G}^\pi(\tilde{s}') \\
(\widetilde{\mathcal{F}}^{\tilde{\pi}}G^\pi)(s_i, s_j) &= \widetilde{R}_{(s_i,s_j)} + \gamma \sum_{(s'_i,s'_j)\in\mathcal{S}^2} \widetilde{T}^{(a_i,a_j)}_{(s_i,s_j)}((s'_i, s'_j))\tilde{\pi}(a_i, a_j|s_i, s_j)G^\pi(s'_i, s'_j) \\
&= |r^\pi_{s_i} - r^\pi_{s_j}| + \gamma \sum_{(s'_i,s'_j)\in\mathcal{S}^2} T^\pi_{s_i}(s'_i)T^\pi_{s_j}(s'_j)G^\pi(s'_i, s'_j) = (\mathcal{F}^\pi_M G^\pi)(s_i, s_j)\,. \square
\end{aligned}
$$

C.5 Expectile Regression

Expectile regression, a method in statistics, is an extension of quantile regression that provides a more detailed analysis of a distribution's tail. This technique aims to estimate the expectiles of a conditional distribution, which are like percentiles but with respect to the mean, not the median. In essence, expectile regression can help capture the structure of data variability and analyze extreme observations in a more precise manner than quantile regression. The $\tau \in (0, 1)$ expectile of some random variable X is defined as a solution to the asymmetric least squares problem:

$$
\arg\min_{m_\tau} \mathbb{E}_{x\sim X}\left[L_2^\tau\left(x - m_\tau\right)\right], \tag{20}
$$

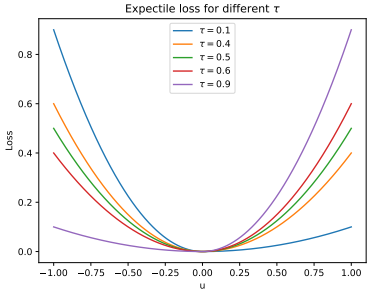

Figure 4: The asymmetric squared loss used for expectile regression. Larger τ gives more weight to positive differences.

where $L_2^\tau(u) = |\tau - \mathbb{1}(u < 0)|u^2$. That is, for $\tau > 0.5$, this asymmetric loss function downweights the contributions of x values smaller than m_τ while giving more weights to larger values. Figure 4 shows the illustration of this asymmetric loss. More detailed descriptions can be found in [25, 33].

D Proof

D.1 Connection between bisimulation error and bisimulation Bellman residual

In this section, we will revise some definitions a bit for obtaining the equivalence between bisimulation error and bisimulation Bellman residual. We first define *bisimulation error* Δ^π_ϕ that measure the distance of the approximation G^π_ϕ to the fixed point $G^\pi_\sim$ as:

$$
\Delta^\pi_\phi := G^\pi_\phi(s_i, s_j) - G^\pi_\sim(s_i, s_j). \tag{21}
$$

And define *bisimulation Bellman residual* ϵ^π_ϕ as:

$$
\epsilon^\pi_\phi := G^\pi_\phi(s_i, s_j) - \mathcal{F}^\pi G^\pi_\phi(s_i, s_j). \tag{22}
$$

Notably, this is slightly different from the notation in Section 4 given the fact that we do not apply absolute value here. Then, we can have the following theorems.

Theorem 14. ***(The bisimulation Bellman residual can be defined as a function of the bisimulation error)***

$$\epsilon^\pi_\phi(s_i, s_j) = \Delta^\pi_\phi(s_i, s_j) - \gamma \mathbb{E}_{\substack{s'_i \sim T^\pi_{s_i} \\ s'_j \sim T^\pi_{s_j}}} [\Delta^\pi_\phi(s'_i, s'_j)], \tag{23}$$

Proof. This follows directly from the bisimulation update operator:

$$\begin{aligned} \epsilon^\pi_\phi(s_i, s_j) &= G^\pi_\phi(s_i, s_j) - \mathcal{F}^\pi G^\pi_\phi(s_i, s_j) \\ &= G^\pi_\sim(s_i, s_j) + \Delta^\pi_\phi(s_i, s_j) - \mathcal{F}^\pi(G^\pi_\sim(s_i, s_j) + \Delta^\pi_\phi(s_i, s_j)) \\ &= \Delta^\pi_\phi(s_i, s_j) - \gamma \mathbb{E}_{\substack{s'_i \sim T^\pi_{s_i} \\ s'_j \sim T^\pi_{s_j}}} [\Delta^\pi_\phi(s'_i, s'_j)] \end{aligned} \tag{24}$$

□

Theorem 15. ***(The bisimulation error can be defined as a function of the bisimulation Bellman residual).*** *For any state pair $(s_i, s_j) \in \mathcal{S} \times \mathcal{S}$, the approximation error $\Delta^\pi_\phi(s_i, s_j)$ can be defined as a function of the Bellman bisimulation error ϵ_ϕ*

$$\Delta^\pi_\phi(s_i, s_j) = \frac{1}{1-\gamma} \mathbb{E}_{(s'_i, s'_j) \sim \mu_\pi} \left[\epsilon^\pi_\phi(s'_i, s'_j) \right]. \tag{25}$$

Proof. Our proof follows similar steps to the proof of Lemma 6.1 in [23] and Theorem 1 in [16]. First by definition:

$$\begin{aligned} \Delta^\pi_\phi(s_i, s_j) &:= G^\pi_\phi(s_i, s_j) - G^\pi_\sim(s_i, s_j) \\ \Rightarrow G^\pi_\sim(s_i, s_j) &= G^\pi_\phi(s_i, s_j) - \Delta^\pi_\phi(s_i, s_j) \end{aligned} \tag{26}$$

Then we can decompose the error:

$$\begin{aligned} \Delta^\pi_\phi(s_i, s_j) &= G^\pi_\phi(s_i, s_j) - G^\pi_\sim(s_i, s_j) \\ &= G^\pi_\phi(s_i, s_j) - \left(|r^\pi_{s_i} - r^\pi_{s_j}| + \gamma \mathbb{E}_{\substack{s'_i \sim T^\pi_{s_i} \\ s'_j \sim T^\pi_{s_j}}} [G^\pi_\sim(s'_i, s'_i)] \right) \\ &= G^\pi_\phi(s_i, s_j) - \left(|r^\pi_{s_i} - r^\pi_{s_j}| + \gamma \mathbb{E}_{\substack{s'_i \sim T^\pi_{s_i} \\ s'_j \sim T^\pi_{s_j}}} [G^\pi_\phi(s'_i, s'_j) - \Delta^\pi_\phi(s'_i, s'_j)] \right) \\ &= G^\pi_\phi(s_i, s_j) - \left(|r^\pi_{s_i} - r^\pi_{s_j}| + \gamma \mathbb{E}_{\substack{s'_i \sim T^\pi_{s_i} \\ s'_j \sim T^\pi_{s_j}}} [G^\pi_\phi(s'_i, s'_j)] \right) + \gamma \mathbb{E}_{\substack{s'_i \sim T^\pi_{s_i} \\ s'_j \sim T^\pi_{s_j}}} \left[\Delta^\pi_\phi(s'_i, s'_j) \right] \\ &= \epsilon^\pi_\phi(s_i, s_j) + \gamma \mathbb{E}_{\substack{s'_i \sim T^\pi_{s_i} \\ s'_j \sim T^\pi_{s_j}}} \left[\Delta^\pi_\phi(s'_i, s'_j) \right] \end{aligned} \tag{27}$$

By considering the operator G as the Bellman evaluation operator for the lifted MDP (See Section C.4), we can rewrite the formula as:

$$\Delta^{\tilde{\pi}}_\phi(\tilde{x}) = \epsilon^{\tilde{\pi}}_\phi(\tilde{x}) + \gamma \mathbb{E}_{\tilde{x}' \sim T^{\tilde{\pi}}_{\tilde{x}}} \left[\Delta^{\tilde{\pi}}_\phi(\tilde{x}') \right]. \tag{28}$$

Then we can treat $\Delta^{\tilde{\pi}}_\phi(\tilde{x})$ as a value function and $\epsilon^{\tilde{\pi}}_\phi(\tilde{x})$ as reward, we can see that:

$$\Delta^{\tilde{\pi}}_\phi(\tilde{x}) = \frac{1}{1-\gamma} \mathbb{E}_{\tilde{x}' \sim T^{\tilde{\pi}}_{\tilde{x}}} \left[\epsilon^{\tilde{\pi}}_\phi(\tilde{x}') \right]. \tag{29}$$

Then we can obtain

$$\Delta^\pi_\phi(s_i, s_j) = \frac{1}{1-\gamma} \mathbb{E}_{(s'_i, s'_j) \sim \mu_\pi} \left[\epsilon^\pi_\phi(s'_i, s'_j) \right]. \tag{30}$$

□

D.2 Thoerem 3

Theorem 3. ***(Bisimulation error upper-bound).*** *Let $\mu_\pi(s)$ denote the stationary distribution over states, let $\mu_\pi(\cdot,\cdot)$ denote the joint distribution over synchronized pairs of states (s_i, s_j) sampled independently from $\mu_\pi(\cdot)$. For any state pair $(s_i, s_j) \in \mathcal{S} \times \mathcal{S}$, the bisimulation error $\Delta^\pi_\phi(s_i, s_j)$ can be upper-bounded by a sum of expected bisimulation Bellman residuals ϵ^π_ϕ:*

$$\Delta^\pi_\phi(s_i, s_j) \leq \frac{1}{1-\gamma}\mathbb{E}_{(s_i,s_j)\sim\mu_\pi}\left[\epsilon^\pi_\phi(s_i, s_j)\right]. \tag{31}$$

Proof. We start from Equation 25 in Section D.1.

$$\begin{aligned}
\Delta^\pi_\phi(s_i, s_j) &= \frac{1}{1-\gamma}\mathbb{E}_{(s'_i,s'_j)\sim\mu_\pi}\left[\epsilon^\pi_\phi(s'_i, s'_j)\right] \\
\Rightarrow |\Delta^\pi_\phi(s_i, s_j)| &= \frac{1}{1-\gamma}\left|\mathbb{E}_{(s'_i,s'_j)\sim\mu_\pi}\left[\epsilon^\pi_\phi(s'_i, s'_j)\right]\right| \\
&\leq \frac{1}{1-\gamma}\mathbb{E}_{(s'_i,s'_j)\sim\mu_\pi}\left[\left|\epsilon^\pi_\phi(s'_i, s'_j)\right|\right].
\end{aligned} \tag{32}$$

Then when we define bisimulation error $\Delta^\pi_\phi(s_i, s_j) := |\Delta^\pi_\phi(s_i, s_j)|$ and bisimulation Bellman residual $\epsilon^\pi_\phi(s'_i, s'_j) := |\epsilon^\pi_\phi(s'_i, s'_j)|$, we have

$$\Delta^\pi_\phi(s_i, s_j) \leq \frac{1}{1-\gamma}\mathbb{E}_{(s'_i,s'_j)\sim\mu_\pi}\left[\epsilon^\pi_\phi(s'_i, s'_j)\right]. \tag{33}$$

□

D.3 Proposition 4

Proposition 4. ***(The expected bisimulation residual is not sufficient over incomplete datasets).*** *If there exists states s'_i and s'_j not contained in dataset $\mathcal{D}$, where the occupancy $\mu_\pi(s'_i|s_i, a_i) > 0$ and $\mu_\pi(s'_j|s_j, a_j) > 0$ for some $s_i \in \mathcal{D}, s_j \in \mathcal{D}$, then there exists a bisimulation measurement and $C > 0$ such that*

- *For all $(\hat{s_i}, \hat{s_j}) \in \mathcal{D}$, the bisimulation Bellman residual $\epsilon^\pi_\phi(\hat{s_i}, \hat{s_j}) = 0$.*
- *There exists $(s_i, s_j) \in \mathcal{D}$, such that the bisimulation error $\Delta^\pi_\phi(s_i, s_j) = C$.*

Proof. This is a direct consequence of Theorem 15. Let $\mathcal{D}'$ contain the set of state pairs (s'_i, s'_j) not contained in the dataset $\mathcal{D}$, where the next-state pair occupancy $\mu_\pi(s'_i, s'_j|s_i, a_i, s_j, a_j) > 0$. Let $\mathcal{D}_{\text{unique}}$ be the set of unique state pairs in $\mathcal{D}$. It follows that

$$\begin{aligned}
\Delta^\pi_\phi(s_i, s_j) =& \frac{1}{1-\gamma}\mathbb{E}_{(s'_i,s'_j)\sim\mu_\pi}\left[\epsilon^\pi_\phi(s'_i, s'_j)\right] \\
=& \frac{1}{1-\gamma}\sum_{(s'_i,s'_j)\sim\mathcal{D}_{\text{unique}}}\mu_\pi((s'_i, s'_j)|s_i, a_i, s_j, a_j)\epsilon^\pi_\phi(s'_i, s'_j) + \\
& \frac{1}{1-\gamma}\sum_{(s'_i,s'_j)\sim\mathcal{D}'}\mu_\pi((s'_i, s'_j)|s_i, a_i, s_j, a_j)\epsilon^\pi_\phi(s'_i, s'_j)
\end{aligned} \tag{34}$$

Recall that $\epsilon^\pi_\phi(s_i, s_j) = \Delta^\pi_\phi(s_i, s_j) - \gamma\mathbb{E}_{\substack{s'_i\sim T^\pi_{s_i} \\ s'_j\sim T^\pi_{s_j}}}[\Delta^\pi_\phi(s'_i, s'_j)]$, and there exists at least one $G(s_i, s_j)$, such that $(s_i, s_j) \in \mathcal{D}'$. Since the sets $\mathcal{D}$ and $\mathcal{D}'$ are distinct, it follows that there exists a measurement G such that $\epsilon^\pi_\phi(s_i, s_j) = 0$ for all $(s_i, s_j) \in \mathcal{D}$, but $\frac{1}{1-\gamma}\sum_{(s'_i,s'_j)\sim\mathcal{D}'}\mu_\pi(s'_i, s'_j|s_i, a_i, s_j, a_j)\epsilon^\pi_\phi(s'_i, s'_j) = C$. □

D.4 Lemma 5

Lemma 5. *For any $\tau \in [0, 1)$, $\mathcal{F}^\pi_\tau$ is a γ_τ-contraction, where $\gamma_\tau = 1 - 2\alpha(1-\gamma)min\{\tau, 1-\tau\}$.*

Proof. Note that $\mathcal{F}^\pi_{1/2}$ is the standard bisimulation operator for π, of which the fixed point is $G^\pi_\sim$. To keep the notation succinct, we will replace G^π with G. For any G_1, G_2,

$$\begin{aligned}
&\mathcal{F}^\pi_{1/2}G_1(s_i,s_j) - \mathcal{F}^\pi_{1/2}G_2(s_i,s_j)\\
&= (G_1(s_i,s_j) + \alpha\mathbb{E}^\pi[\delta_i]) - (G_2(s_i,s_j) + \alpha\mathbb{E}^\pi[\delta_j])\\
&= (1-\alpha)(G_1(s_i,s_j) - G_2(s_i,s_j)) + \alpha\mathbb{E}_\pi[(1-\gamma)|r^\pi_{s_i} - r^\pi_{s_j}| + \gamma G_1(s'_i,s'_j)\\
&\qquad\qquad - (1-\gamma)|r^\pi_{s_i} - r^\pi_{s_j}| - \gamma G_2(s'_i,s'_j)] \qquad (35)\\
&= (1-\alpha)(G_1(s_i,s_j) - G_2(s_i,s_j)) + \alpha\mathbb{E}_\pi[\gamma G_1(s'_i,s'_j) - \gamma G_2(s'_i,s'_j)]\\
&\le (1-\alpha)\|G_1 - G_2\|_\infty + \alpha\gamma\|G_1 - G_2\|_\infty\\
&= (1-\alpha(1-\gamma))\|G_1 - G_2\|_\infty.
\end{aligned}$$

When $\tau \neq \frac{1}{2}$, we introduce two more operators to simplify the analysis:

$$\begin{aligned}
(\mathcal{F}^\pi_+ G_1)(s_i,s_j) &= G(s_i,s_j) + \mathbb{E}^\pi[\delta]_+\\
(\mathcal{F}^\pi_- G_2)(s_i,s_j) &= G(s_i,s_j) + \mathbb{E}^\pi[\delta]_-
\end{aligned} \qquad (36)$$

Now we show that both operators meet the Banach-fixed point theorem (e.g. $\|\mathcal{F}^\pi_+ G_1 - \mathcal{F}^\pi_+ G_2\|_\infty \le \|G_1 - G_2\|_\infty$). For any G_1, G_2:

$$\begin{aligned}
&(\mathcal{F}^\pi_+ G_1)(s_i,s_j) - (\mathcal{F}^\pi_+ G_2)(s_i,s_j)\\
&= G_1 - G_2 + \mathbb{E}^\pi[[\delta_i]_+ - [\delta_j]_+]\\
&= \mathbb{E}^\pi[G_1 + [\delta_i]_+ - (G_2 + [\delta_j]_+)]
\end{aligned} \qquad (37)$$

The relationship between $G_1 + [\delta_i]_+$ and $G_2 + [\delta_j]_+$ exists in four cases:

- $\delta_i \ge 0, \delta_j \ge 0$, then

$$G_1 + [\delta_i]_+ - (G_2 + [\delta_j]_+) = \gamma(G_1(s'_i,s'_j) - G_2(s'_i,s'_j)). \qquad (38)$$

- $\delta_i < 0, \delta_j < 0$, then

$$G_1 + [\delta_i]_+ - (G_2 + [\delta_j]_+) = G_1(s_i,s_j) - G_2(s_i,s_j). \qquad (39)$$

- $\delta_i \ge 0, \delta_j < 0$, then

$$\begin{aligned}
&G_1 + [\delta_i]_+ - (G_2 + [\delta_j]_+)\\
&= (1-\gamma)|r^\pi_{s_i} - r^\pi_{s_j}| + \gamma G_1(s'_i,s'_j) - G_2(s_i,s_j)\\
&< (1-\gamma)|r^\pi_{s_i} - r^\pi_{s_j}| + \gamma G_1(s'_i,s'_j) - ((1-\gamma)|r^\pi_{s_i} - r^\pi_{s_j}| + \gamma G_2(s'_i,s'_j))\\
&= \gamma(G_1(s'_i,s'_j) - G_2(s'_i,s'_j)),
\end{aligned} \qquad (40)$$

  where the inequality comes from $G_2(s_i,s_j) > (1-\gamma)|r^\pi_{s_i} - r^\pi_{s_j}| + \gamma G_2(s'_i,s'_j)$.

- $\delta_i < 0, \delta_j \ge 0$, then

$$\begin{aligned}
&G_1 + [\delta_i]_+ - (G_2 + [\delta_j]_+)\\
&= G_1(s_i,s_j) - ((1-\gamma)|r^\pi_{s_i} - r^\pi_{s_j}| + G_2(s'_i,s'_j))\\
&\le G_1(s_i,s_j) - G_2(s_i,s_j),
\end{aligned} \qquad (41)$$

  where the inequality comes from $G_2(s_i,s_j) \le (1-\gamma)|r^\pi_{s_i} - r^\pi_{s_j}| + \gamma G_2(s'_i,s'_j)$.

As a result, we have $(\mathcal{F}_+^\pi G_1)(s_i,s_j) - (\mathcal{F}_+^\pi G_2)(s_i,s_j) \le \|G_1 - G_2\|_\infty$. Combine $\mathcal{F}_+^\pi$ and $\mathcal{F}_-^\pi$, we can rewrite $\mathcal{F}_\tau^\pi$ as:

$$\begin{aligned}\mathcal{F}_\tau^\pi G(s_i,s_j) &= G(s_i,s_j) + 2\alpha\mathbb{E}^\pi[\tau[\delta]_+ + (1-\tau)[\delta]_-] \\ &= (1-2\alpha)G(s_i,s_j) + 2\alpha\tau(G(s_i,s_j) + \mathbb{E}^\pi[[\delta]_+] + 2\alpha(1-\tau)(G(s_i,s_j) + \mathbb{E}^\pi[[\delta]_-]) \\ &= (1-2\alpha)G(s_i,s_j) + 2\alpha\tau(\mathcal{F}_+^\pi G_1)(s_i,s_j) + 2\alpha(1-\tau)(\mathcal{F}_-^\pi G_1)(s_i,s_j).\end{aligned} \tag{42}$$

What's more

$$\begin{aligned}\mathcal{F}_{\frac{1}{2}}^\pi G(s_i,s_j) &= G(s_i,s_j) + \alpha\mathbb{E}^\pi[\delta] \\ &= G(s_i,s_j) + \alpha((\mathcal{F}_+^\pi G_1)(s_i,s_j) + (\mathcal{F}_-^\pi G_1)(s_i,s_j) - 2\alpha G(s_i,s_j)) \\ &= (1-2\alpha)G(s_i,s_j) + \alpha((\mathcal{F}_+^\pi G_1)(s_i,s_j) + (\mathcal{F}_-^\pi G_1)(s_i,s_j)).\end{aligned} \tag{43}$$

When $\tau > \frac{1}{2}$, for any G_1 and G_2:

$$\begin{aligned}&(\mathcal{F}_\tau^\pi G_1)(s_i,s_j) - (\mathcal{F}_\tau^\pi G_2)(s_i,s_j) \\ &= (1-2\alpha)(G_1(s_i,s_j) - G_2(s_i,s_j)) + 2\alpha\tau((\mathcal{F}_+^\pi G_1)(s_i,s_j) - (\mathcal{F}_+^\pi G_2)(s_i,s_j)) \\ &\qquad + 2\alpha(1-\tau)((\mathcal{F}_-^\pi G_1)(s_i,s_j) - (\mathcal{F}_-^\pi G_2)(s_i,s_j)) \\ &= (1-2\alpha-2(1-2\alpha)(1-\tau))(G_1(s_i,s_j) - G_2(s_i,s_j))_2(1-\tau)((\mathcal{F}_{\frac{1}{2}}^\pi G_1)(s_i,s_j) - (\mathcal{F}_{\frac{1}{2}}^\pi G_2)(s_i,s_j)) \\ &\qquad - 2\alpha(1-2\tau)((\mathcal{F}_+^\pi G_1)(s_i,s_j) - (\mathcal{F}_+^\pi G_2)(s_i,s_j)) \\ &\le (1-2\alpha-2(1-2\alpha)(1-\tau))\|G_1(s_i,s_j) - G_2(s_i,s_j)\|_\infty \\ &\qquad + 2(1-\tau)(1-\alpha(1-\gamma))\|G_1(s_i,s_j) - G_2(s_i,s_j)\|_\infty \\ &\qquad - 2\alpha(1-2\tau)\|G_1(s_i,s_j) - G_2(s_i,s_j)\|_\infty \\ &= (1-2\alpha(1-\tau)(1-\gamma))\|G_1(s_i,s_j) - G_2(s_i,s_j)\|_\infty\end{aligned} \tag{44}$$

When $\tau < \frac{1}{2}$, for any G_1 and G_2:

$$\begin{aligned}&(\mathcal{F}_\tau^\pi G_1)(s_i,s_j) - (\mathcal{F}_\tau^\pi G_2)(s_i,s_j) \\ &= (1-2\alpha)(G_1(s_i,s_j) - G_2(s_i,s_j)) + 2\alpha\tau((\mathcal{F}_+^\pi G_1)(s_i,s_j) - (\mathcal{F}_+^\pi G_2)(s_i,s_j)) \\ &\qquad + 2\alpha(1-\tau)((\mathcal{F}_-^\pi G_1)(s_i,s_j) - (\mathcal{F}_-^\pi G_2)(s_i,s_j)) \\ &= (1-2\alpha-2\tau(1-2\alpha))(G_1(s_i,s_j) - G_2(s_i,s_j)) + 2\tau((\mathcal{F}_{\frac{1}{2}}^\pi G_1)(s_i,s_j) - (\mathcal{F}_{\frac{1}{2}}^\pi G_2)(s_i,s_j)) \\ &\qquad + 2\alpha(1-2\tau)((\mathcal{F}_-^\pi G_1)(s_i,s_j) - (\mathcal{F}_-^\pi G_2)(s_i,s_j)) \\ &\le (1-2\alpha-2\tau(1-2\alpha))\|G_1(s_i,s_j) - G_2(s_i,s_j)\|_\infty \\ &\qquad + 2\tau(1-\alpha(1-\gamma))\|G_1(s_i,s_j) - G_2(s_i,s_j)\|_\infty \\ &\qquad + 2\alpha(1-2\tau)\|G_1(s_i,s_j) - G_2(s_i,s_j)\|_\infty \\ &= (1-2\alpha\tau(1-\gamma))\|G_1(s_i,s_j) - G_2(s_i,s_j)\|_\infty.\end{aligned} \tag{45}$$

□

D.5 Lemma 6

Lemma 6. *For any $\tau, \tau' \in [0,1)$ with $\tau' \ge \tau$, and for all $s_i, s_j \in \mathcal{S}$ and any α, we have $G_{\tau'} \ge G_\tau$.*

Proof. We denote $G_{\tau'}$ is the fixed point of applying the operator $\mathcal{F}_{\tau'}^\pi$, and G_τ is the fixed point of applying the operator $\mathcal{F}_\tau$. Based on Equation 6, we have:

$$\begin{aligned}&\mathcal{F}_{\tau'}^\pi G(s_i,s_j) - \mathcal{F}_\tau^\pi G(s_i,s_j) \\ &= (1-2\alpha)G(s_i,s_j) + 2\alpha\tau'\mathcal{F}_+^\pi G(s_i,s_j) + 2\alpha(1-\tau')\mathcal{F}_-^\pi G(s_i,s_j) \\ &\qquad - ((1-2\alpha)G(s_i,s_j) + 2\alpha\tau\mathcal{F}_+^\pi G(s_i,s_j) + 2\alpha(1-\tau)\mathcal{F}_-^\pi G(s_i,s_j)) \\ &= 2\alpha(\tau'-\tau)(\mathcal{F}_+^\pi G(s_i,s_j) - \mathcal{F}_-^\pi G(s_i,s_j)) \\ &= 2\alpha(\tau'-\tau)\mathbb{E}^\pi[[\delta]_+ - [\delta]_-] \ge 0.\end{aligned} \tag{46}$$

Therefore $G_{\tau'} > G_\tau$. □

D.6 Theorem 7

Theorem 7. *In deterministic MDP and fixed finite dataset, we have:*

$$\lim_{\tau \to 1} G_\tau(s_i, s_j) = \max_{\substack{a_i \in \mathcal{A}, a_j \in \mathcal{A} \\ \text{s.t. } \pi_\beta(a_i|s_i)>0, \pi_\beta(a_j|s_j)>0}} G^*_\sim((s_i, a_i), (s_j, a_j)). \tag{47}$$

where $G^*_\sim((s_i, a_i), (s_j, a_j))$ is a fixed-point measurement constrained to the dataset and defined on state-action space $\mathcal{S} \times \mathcal{A}$ as

$$G^*_\sim((s_i, a_i), (s_j, a_j)) = |r(s_i, a_i) - r(s_j, a_j)| + \gamma \mathbb{E}_{\substack{s'_i \sim T^{\pi_\beta}_{s_i} \\ s'_j \sim T^{\pi_\beta}_{s_j}}} \left[\max_{\substack{a'_i \in \mathcal{A}, a'_j \in \mathcal{A} \\ \text{s.t. } \pi_\beta(a'_i|s'_i)>0, \pi_\beta(a'_j|s'_j)>0}} G^*_\sim((s'_i, a'_i), (s'_j, a'_j)) \right]. \tag{48}$$

Proof. First, we can easily proof that $G^*_\sim(s_i, a_i, s_j, a_j)$ is a fixed point. Define the corresponding operator of $G^*_\sim$ is F^*, we can know that F^* is a contraction. Then, we have

Corollary 16. *For any τ, $s_i, s_j \in \mathcal{S}$ we have*

$$G_\tau(s_i, s_j) \leq \max_{\substack{a_i \in \mathcal{A}, a_j \in \mathcal{A} \\ \text{s.t. } \pi_\beta(a_i|s_i)>0, \pi_\beta(a_j|s_j)>0}} G^*_\sim((s_i, a_i), (s_j, a_j)) \tag{49}$$

Proof. The proof follows from the observation that convex combination is smaller than maximum. □

Besides, we also have

Lemma 17. *Let X be a real-valued random variable with a bounded support and supremum of the support is x^*. Then,*

$$\lim_{\tau \to 1} m_\tau = x^*$$

Proof. Same as the Lemma 1 in [25]. One can show that expectiles of a random variable have the same supremum x^*. Moreover, for all τ_1 and τ_2 such that $\tau_1 < \tau_2$, we get $m_{\tau_1} \leq m_{\tau_2}$. Therefore, the limit follows from the properties of bounded monotonically non-decreasing functions. □

Combining Corollary 16 and Lemma 17, we can obtain the above.

□

D.7 Theorem 8

Theorem 8. *(Value bound based on on-policy bisimulation measurements in terms of encoder error). Given an MDP $\widetilde{\mathcal{M}}$ constructed by aggregating states in an ω-neighborhood, and an encoder ϕ that maps from states in the original MDP $\mathcal{M}$ to these clusters, the value functions for the two MDPs are bounded as*

$$\left| V^\pi(s_i) - \widetilde{V}^\pi(\phi(s_i)) \right| \leq \frac{2\omega + \hat{\Delta}}{c_r(1-\gamma)}. \tag{50}$$

where $\hat{\Delta} := \|\hat{G}^\pi_\sim - \hat{G}^\pi_\phi\|_\infty$ is the approximation error.

Proof. Let the reward function be bounded as $R \in [0, 1]$, $\phi : \mathcal{S} \to \widetilde{\mathcal{S}}$, and $\phi(s_i) = \phi(s_j) \Rightarrow \hat{G}^\pi_\phi(s_i, s_j) = |\phi(s_i) - \phi(s_j)| \leq 2\,\omega$, we can conduct an aggregat MDP $\widetilde{M} = (\widetilde{\mathcal{S}}, \mathcal{A}, \widetilde{T}, \widetilde{R})$. Let ξ be

a measure on $\mathcal{S}$. Following Lemma 8 in [24], we have that:

$$\begin{aligned}\left|V^{\pi}\left(s_i\right)-\widetilde{V}^{\pi}\left(\phi\left(s_i\right)\right)\right| &\leq \frac{c_r^{-1}}{\xi(\phi(s))}\int\limits_{z\in\phi(s)} c_R\left|r^{\pi}(s)-r^{\pi}(z)\right| \\ &\quad+(1-\gamma)\left|\int\limits_{s'\in\mathcal{S}}(T^{\pi}(s'|s)-T^{\pi}(s'|z))\frac{c_r\gamma}{1-\gamma}V^{\pi}(s')ds'\right|d\xi(z)+\gamma\|V^{\pi}-\widetilde{V}^{\pi}\|_{\infty} \\ &\leq \frac{c_r^{-1}}{\xi(\phi(s))}\int\limits_{z\in\phi(s)} G_{\sim}^{\pi}(s,z)d\xi(z)+\gamma\|V^{\pi}-\widetilde{V}^{\pi}\|_{\infty}\end{aligned} \tag{51}$$

Thus, taking the supremum on the LHS, we have:

$$\begin{aligned}(1-\gamma)\left|V^{\pi}\left(s_i\right)-\widetilde{V}^{\pi}\left(\phi\left(s_i\right)\right)\right| &\leq \frac{c_r^{-1}}{\xi(\phi(s))}\int\limits_{z\in\phi(s)} G_{\sim}^{\pi}(s,z)d\xi(z) \\ &\leq \frac{c_r^{-1}}{\xi(\phi(s))}\int\limits_{z\in\phi(s)} \hat{G}_{\phi}^{\pi}(s,z)+\|G_{\sim}^{\pi}-\hat{G}_{\phi}^{\pi}\|_{\infty}d\xi(z) \\ &= \frac{c_r^{-1}}{\xi(\phi(s))}\int\limits_{z\in\phi(s)} (2\omega+\hat{\Delta})d\xi(z) \\ &= c_r^{-1}(2\omega+\hat{\Delta}).\end{aligned} \tag{52}$$

Therefore,

$$\left|V^{\pi}\left(s_i\right)-\widetilde{V}^{\pi}\left(\phi\left(s_i\right)\right)\right| \leq \frac{2\omega+\hat{\Delta}}{c_r(1-\gamma)}, \tag{53}$$

□

E Understanding of Theorem 7

Theorem 7. *In deterministic MDP and fixed finite dataset, we have:*

$$\lim_{\tau\to 1} G_{\tau}(s_i,s_j) = \max_{\substack{a_i\in\mathcal{A},a_j\in\mathcal{A}\\ s.t.\ \pi_{\beta}(a_i|s_i)>0,\pi_{\beta}(a_j|s_j)>0}} G_{\sim}^{*}((s_i,a_i),(s_j,a_j)). \tag{54}$$

where $G_{\sim}^{*}((s_i,a_i),(s_j,a_j))$ is a fixed-point measurement constrained to the dataset and defined on the state-action space $\mathcal{S}\times\mathcal{A}$ as

$$G_{\sim}^{*}((s_i,a_i),(s_j,a_j)) = |r(s_i,a_i)-r(s_j,a_j)|+\gamma\mathbb{E}_{\substack{s_i'\sim T_{s_i}^{\pi_\beta}\\ s_j'\sim T_{s_j}^{\pi_\beta}}}\left[\max_{\substack{a_i'\in\mathcal{A},a_j'\in\mathcal{A}\\ s.t.\ \pi_{\beta}(a_i'|s_i')>0,\pi_{\beta}(a_j'|s_j')>0}} G_{\sim}^{*}((s_i',a_i'),(s_j',a_j'))\right]. \tag{55}$$

Given the MDP specified by the tuple $(\mathcal{S},\mathcal{A},T,R)$, we construct a lifted MDP $(\widetilde{\mathcal{S}},\widetilde{\mathcal{A}},\widetilde{T},\widetilde{R})$, by taking the state space to be $\widetilde{\mathcal{S}}=\mathcal{S}^2$, the action space to be $\widetilde{\mathcal{A}}=\mathcal{A}^2$, the transition dynamics to be given by $\widetilde{T}_{\tilde{s}}^{\tilde{a}}(\tilde{s}')=\widetilde{T}_{(s_i,s_j)}^{(a_i,a_j)}((s_i',s_j'))=T_{s_i}^{a_i}(s_i')T_{s_j}^{a_j}(s_j')$ for all $(s_i,s_j),(s_i',s_j')\in\mathcal{S}^2$, $a_i,a_j\in\mathcal{A}$, and the action-independent rewards to be $\widetilde{R}_{\tilde{s}}=\widetilde{R}_{(s_i,s_j)}=|r_{s_i}^{\pi}-r_{s_j}^{\pi}|$ for all $s_i,s_j\in\mathcal{S}$. The Bellman evaluation operator $\widetilde{\mathcal{F}}^{\tilde{\pi}}$ for this lifted MDP at discount rate γ under the policy $\tilde{\pi}(\tilde{a}|\tilde{s})=\tilde{\pi}(a_i,a_j|s_i,s_j)=\pi(a_i|s_i)\pi(a_j|s_j)$ is given by (for all $G^{\pi}\in\mathbb{R}^{\mathcal{S}\times\mathcal{S}}$ and $(s_i,s_j)\in\mathcal{S}\times\mathcal{S}$):

$$(\widetilde{\mathcal{F}}^{\tilde{\pi}}Q^{*})(\tilde{s},\tilde{a}) = \widetilde{R}_{\tilde{s},\tilde{a}}+\gamma\sum_{\tilde{s}\in\widetilde{\mathcal{S}}}\widetilde{T}_{\tilde{s}}^{\tilde{a}}(\tilde{s}')\max_{\tilde{a}\in\widetilde{A}}Q^{*}(\tilde{s}',\tilde{a}'). \tag{56}$$

Though similar, Equation 55 has more constraints as it requires the possibility of $\pi_\beta(a'_i|s'_i)$ and $\pi_\beta(a'_j|s'_j)$ are larger than zero in the dataset. As such, we may also change the Equation 56 to:

$$(\widetilde{\mathcal{F}}^{\tilde{\pi}}Q^*)(\tilde{s},\tilde{a}) = \widetilde{R}_{\tilde{s},\tilde{a}} + \gamma \sum_{\tilde{s}\in\widetilde{\mathcal{S}}} \widetilde{T}^{\tilde{a}}_{\tilde{s}}(\tilde{s}') \max_{\substack{\tilde{a}'\in\mathcal{A} \\ \text{s.t. } \tilde{\pi}_\beta(\tilde{a}'|\tilde{s}')>0}} Q^*(\tilde{s}',\tilde{a}'). \tag{57}$$

This is, indeed, equivalent to the *in-sample*-style Q function in [25]. Intuitively, $G^*_\sim((s_i,a_i),(s_j,a_j))$ can be interpreted as the optimal state-action value function $Q^*(\tilde{s},\tilde{a})$ in a lifted MDP $\widetilde{M}$. Then $G^\pi_\sim((s_i,a_i),(s_j,a_j))$ is the state-action value function $Q^\pi(\tilde{s},\tilde{a})$ that associated with policy π, and $G_\sim(s_i,s_j)$ as a state value function $V(\tilde{s})$. And therefore, we can connect our expectile-based bisimulation operator to the lifted MDP, where we can use the conventional analytics tools in RL to analyze bisimulation operators.

F Additional Experiments

F.1 Ablation Study

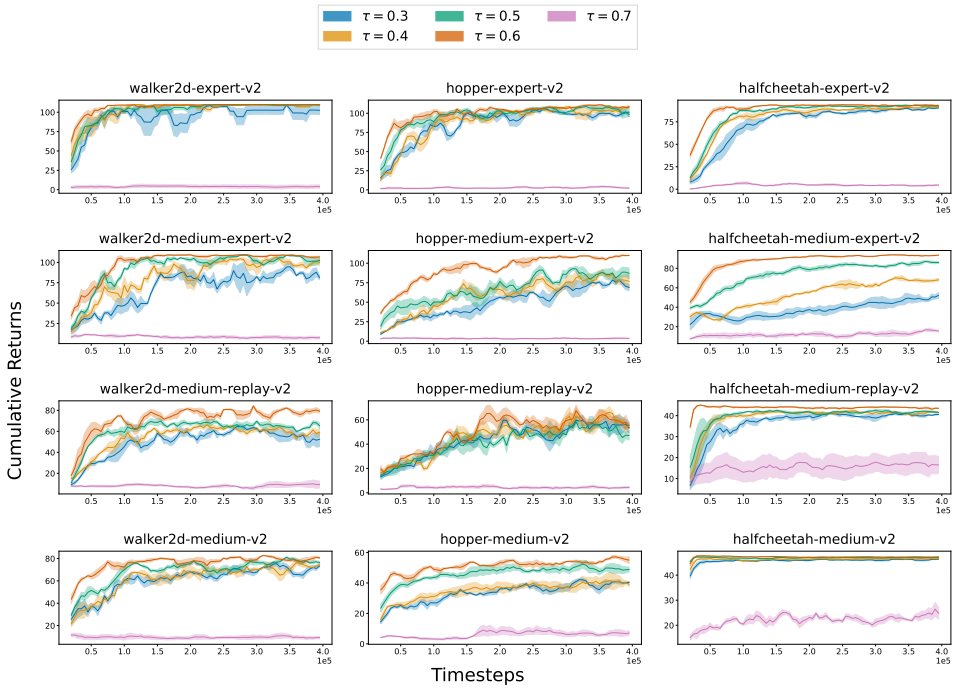

Figure 5: Performance comparison on 12 D4RL tasks over 10 seeds with one standard error shaded in the default setting.

Here we present the ablation study of setting different expectile $\tau \in \{0.3, 0.4, \cdots, 0.7\}$ in Figure 5 to investigate the effect of the critical hyper-parameter in EBS. The experimental results demonstrate that the final performance gradually improves with a larger τ. Notably, the most superior performance is achieved when τ equals 0.6. However, when τ further increases to 0.7, the agent's performance suffers a sharp decline. We hypothesize that this could be due to the value function possibly exploding when τ is set to larger values, subsequently leading to poorer performance outcomes. This is as expected since the over-large τ leads to the overestimation error caused by neural networks. The experimental results demonstrate that we can balance a trade-off between minimizing the expected bisimulation residual and evaluating "optimal" measurement solely on the dataset by choosing a suitable τ.

G Additional Related Works

Here we present a brief introduction of all baselines we used in the experiments:

TD3BC [15] add a behavior cloning term to regularize the policy of the TD3 [17] algorithm, achieves a state-of-the-art performance in Offline settings.

DrQ+BC [32] combining data augmentation techniques with the TD3+BC method, which applies TD3 in the offline setting with a regularizing behavioral-cloning term to the policy loss. The policy objective is: $\pi = \underset{\pi}{\operatorname{argmax}} \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[\lambda Q(s, \pi(s)) - (\pi(s) - a)^2\right]$

DRIML [35] and HOMER [36] (Time Contrastive methods) learn representations which can discriminate between adjacent observations in a rollout and pairs of random observations.

CURL [28] (Augmentation Contrastive method) learns a representation that is invariant to a class of data augmentations while being different across random example pairs.

Inverse Model [40] (One-Step Inverse Models) predict the action taken conditioned on the previous and resulting observations.