STABLE OFFLINE VALUE FUNCTION LEARNING WITH BISIMULATION-BASED REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

In reinforcement learning, offline value function learning is the procedure of using an offline dataset to estimate the expected discounted return from each state when taking actions according to a fixed target policy. The stability of this procedure, i.e., whether it converges to its fixed-point, critically depends on the representations of the state-action pairs. Poorly learned representations can make value function learning unstable, or even divergent. Therefore, it is critical to stabilize value function learning by explicitly shaping the state-action representations. Recently, the class of bisimulation-based algorithms have shown promise in shaping representations for control. However, it is still unclear if this class of methods can *stabilize* value function learning. In this work, we investigate this question and answer it affirmatively. We introduce a bisimulation-based algorithm called kernel representations for offline policy evaluation (KROPE). KROPE uses a kernel to shape state-action representations such that state-action pairs that have similar immediate rewards and lead to similar next state-action pairs under the target policy also have similar representations. We show that KROPE: 1) learns stable representations and 2) leads to lower value error than baselines. Our analysis provides new theoretical insight into the stability properties of bisimulation-based methods and suggests that practitioners can use these methods for stable and accurate evaluation of offline reinforcement learning agents.

1 INTRODUCTION

Learning the value function of a policy is a critical component of many reinforcement learning (RL) algorithms (Sutton & Barto, 2018). While value function learning algorithms such as temporal-difference learning (TD) have been successful, they can be unreliable. In particular, the deadly triad, i.e., the combination of off-policy updates, function approximation, and bootstrapping, can make TD-based methods diverge (Sutton & Barto, 2018; Tsitsiklis & Van Roy, 1997; Baird, 1995). Function approximation is a critical component of value function learning since it determines the representations of state-action pairs, which in turn defines the space of expressible value functions. Depending on how this value function space is represented, value function learning algorithms may diverge (Ghosh & Bellemare, 2020). That is, the value function learning algorithm may not converge to its fixed-point, or may even diverge away from it. In this work, we investigate how to explicitly learn state-action representations to stabilize value function learning.

In seeking such representations, we turn to π-bisimulation algorithms. These algorithms define a metric to capture *behavioral similarity* between state-action pairs such that similarity is based on immediate rewards received and the similarity of next state-action pairs visited by π (Castro, 2019). The algorithms then use this metric to learn representations such that state-action pairs that are similar under this metric have similar representations. Ultimately, the goal of π-bisimulation methods is to learn representations such that state-actions pairs with similar values under π also have similar representations (see Figure 1). While these algorithms have shown promise in improving the expected return of RL algorithms, it remains unclear whether they contribute to stability (Castro et al., 2023; Zhang et al., 2021; Castro et al., 2022). In this paper, we aim to understand whether π-bisimulation-based representations stabilize value function learning.

In this work, we focus on offline value function learning. Given a fixed, offline dataset generated by unknown and possibly multiple behavior policies, the goal is to estimate the value function of

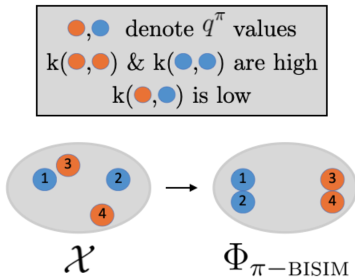


Figure 1: The figure illustrates the native state-action representations $\mathcal{X}$ and π-bisimulation representations $\Phi_{\pi-\text{BISIM}}$. π-bisimulation algorithms use a similarity function k that captures similarity between state-action pairs based on immediate rewards and similarity of next state-action pairs under π to shape their representations. Ultimately, the goal of π-bisimulation methods is to learn representations such that state-actions pairs with similar values under π also have similar representations. The function k outputs high values within the blue (and orange) state-actions but low values between blue and orange state-actions. Therefore, the blue (and orange) state-actions have similar representations, but different representations between the distinct colors.

a fixed, target policy. We introduce kernel representations for offline policy evaluation (KROPE), a bisimulation-based representation learning algorithm. KROPE defines a kernel that captures similarity between state-action pairs based on immediate rewards received and similarity of next state-action pairs under the *target* policy. It then shapes the state-action representations such that state-action pairs that are similar according to this kernel have similar representations. We use KROPE as the representative algorithm for the class of bisimulation-based representation learning algorithms to investigate the following question:

Can bisimulation-based representation learning stabilize offline value function learning?

Through theoretical and empirical analysis, we answer this question affirmatively and make the following contributions:

1. We introduce kernel representations for offline policy evaluation (KROPE) for stable and accurate offline value function learning (Section 3).
2. We prove that KROPE's representations stabilize least-squares policy evaluation (LSPE), a popular value function learning algorithm (Sections 3.2).
3. We prove that KROPE representations are Bellman complete, another indication of stability (Sections 3.3).
4. We empirically validate that KROPE representations lead to more stable and accurate offline value function learning compared to non-bisimulation baselines (Section 4).
5. We empirically analyze the sensitivity of the KROPE learning procedure under the deadly triad. These experiments shed light on when representation pre-training may be easier than direct value function learning with LSPE (Appendix C.3.1).

2 Background

In this section, we present our problem setup and discuss prior work.

2.1 Problem Setup and Notation

We consider the infinite-horizon Markov decision process (MDP) framework (Puterman, 2014), $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, r, P, \gamma, d_0 \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $r : \mathcal{S} \times \mathcal{A} \to [-1, 1]$ is the deterministic reward function, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition dynamics function, $\gamma \in [0, 1)$ is the discount factor, and $d_0 \in \Delta(\mathcal{S})$ is the initial state distribution, where $\Delta(X)$ represents the set of all probability distributions over a set X. We refer to the joint state-action space as $\mathcal{X} := \mathcal{S} \times \mathcal{A}$. The agent acting according to policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ in the MDP generates a trajectory: $S_0, A_0, R_0, S_1, A_1, R_1, ...$, where $S_0 \sim d_0$, $A_t \sim \pi(\cdot|S_t)$, $R_t := r(S_t, A_t)$, and $S_{t+1} \sim P(\cdot|S_t, A_t)$ for $t \geq 0$.

We define the action-value function of a policy π for a given state-action pair as $q^\pi(s, a) := \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t r(S_t, A_t)|S_0 = s, A_0 = a]$, i.e., the expected discounted return when starting from state s with initial action a and then following policy π. The Bellman evaluation operator $\mathcal{T}^\pi : \mathbb{R}^{\mathcal{X}} \to \mathbb{R}^{\mathcal{X}}$

is defined as $(\mathcal{T}^\pi f)(s,a) := r(s,a) + \gamma \mathbb{E}_{S' \sim P(\cdot|s,a), A' \sim \pi}[f(S',A')], \forall f \in \mathbb{R}^{\mathcal{X}}$. Accordingly, the action-value function satisfies the Bellman equation, i.e., $q(s,a) = r(s,a) + \gamma \mathbb{E}_{P,\pi_e}[q(S',A')]$.

It will be convenient to consider the matrix notation equivalents of the above functions. Since a policy π induces a Markov chain on $\mathcal{X}$, we can denote the transition matrix of this Markov chain by $P^\pi \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{X}|}$. Here, each entry $P^\pi(i,j)$ is the probability of transitioning from state-actions i to j. Similarly, we have the action-value function $q^\pi \in \mathbb{R}^{|\mathcal{X}|}$ and reward vector $r \in \mathbb{R}^{|\mathcal{X}|}$, where the entry $q^\pi(i)$ and $r(i)$ are the expected discounted return from state-action i under π and reward received at state-action i respectively.

In this work, we study the representations of the state-action space. We use $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ to denote the state-action representations, which maps state-action pairs into a d-dimensional Euclidean space. We denote the matrix of all the state-action features as $\Phi \in \mathbb{R}^{|\mathcal{X}| \times d}$, where each row is the state-action feature $\phi(s,a) \in \mathbb{R}^d$ for state-action pair (s,a). When dealing with the offline dataset $\mathcal{D}$, Φ's dimensions are $|\mathcal{D}| \times\ d$, where $|\mathcal{D}|$ is the number state-actions in the dataset $\mathcal{D}$. Note that Φ can be the native state-action features of the MDP, or the output of some representation learning algorithm, or the penultimate features of the action-value function when using a neural network. Throughout this paper, we will view ϕ as an encoder or state-action abstraction (Li et al., 2006). Note that the state-action abstraction view enables us to view ϕ as a state-action aggregator from the space of state-actions $\mathcal{X}$ to the space of state-action groups $\mathcal{X}^\phi$.

2.2 Offline Policy Evaluation and Value Function Learning

In offline policy evaluation (OPE), the goal is to evaluate a fixed target policy, π_e, using a fixed dataset of m transition tuples $\mathcal{D} := \{(s_i, a_i, s_i', r_i)\}_{i=1}^m$. In this work, we evaluate π_e by estimating the *action-value function* q^{π_e} using $\mathcal{D}$. Crucially, $\mathcal{D}$ may have been generated by a set of *unknown behavior* policies that are different from π_e, which means that simply averaging the discounted returns in $\mathcal{D}$ will produce an inconsistent estimate of q^{π_e}. In our theoretical results, we make the standard coverage assumption that $\forall s \in \mathcal{S}, \forall a \in \mathcal{A}$ if $\pi_e(a|s) > 0$, then the state-action pair (s,a) has non-zero probability of appearing in $\mathcal{D}$ (Sutton & Barto, 2018; Precup et al., 2000).

We measure the accuracy of the value function estimate with the *mean squared value error* (MSVE). Let $\hat{q}^{\pi_e}$ be the estimate returned by a value function learning method using $\mathcal{D}$. The MSVE of this estimate is defined as $\text{MSVE}[\hat{q}^{\pi_e}] := \mathbb{E}_{(S,A)\sim\mathcal{D}}[(\hat{q}^{\pi_e}(S,A) - q^{\pi_e}(S,A))^2]$. In environments with continuous state-action spaces, where it is impossible to compute q^{π_e} for all state-actions, we adopt a common evaluation procedure from the OPE literature of measuring the MSE across only the initial state-action distribution, i.e., $\text{MSE}[\hat{q}^{\pi_e}] := \mathbb{E}_{S_0 \sim d_0, A_0 \sim \pi_e}[(\hat{q}^{\pi_e}(S_0, A_0) - q^{\pi_e}(S_0,A_0))^2]$. For this procedure, we assume access to d_0 (Voloshin et al., 2021; Fu et al., 2021). While in practice q^{π_e} is unknown, it is standard for the sake of empirical analysis to estimate q^{π_e} by executing unbiased Monte Carlo rollouts of π_e or computing q^{π_e} exactly using dynamic programming in tabular environments (Voloshin et al., 2021; Fu et al., 2021).

Least-Squares Policy Evaluation Least-squares policy evaluation (LSPE) is a value function learning algorithm, which models the action-value function as a linear function: $\hat{q}_\theta^{\pi_e}(s,a) := \phi(s,a)^\top \theta$, where $\theta \in \mathbb{R}^d$ (Nedic & Bertsekas, 2003). LSPE iteratively learns θ with the following updates per iteration step t:

$$\theta_{t+1} \leftarrow (\mathbb{E}_{\mathcal{D}}[\Phi^\top \Phi])^{-1} \mathbb{E}_{\mathcal{D},\pi_e}[\Phi^\top (r + \gamma P^{\pi_e} \Phi \theta_t)], \tag{1}$$

where the expectations are taken with respect to the randomness of the dataset $\mathcal{D}$ and π_e. Note that $\mathbb{E}[\Phi^\top \Phi]$ is the feature covariance matrix. Assuming LSPE converges, it will converge to the same fixed-point as TD(0) (Szepesvari, 2010), which we denote as θ_{LSPE}. In this work, we follow a two-stage approach to applying LSPE: we first obtain the encoder ϕ either through representation learning or using the native features of the MDP, and then feed the obtained ϕ along with $\mathcal{D}$ and π_e as input to LSPE, which outputs $\hat{q}_\theta^{\pi_e}$ (Nedic & Bertsekas, 2003; Chang et al., 2022). This two-stage approach of learning a linear function on top of fixed representations is called the linear evaluation protocol (Chang et al., 2022; Farebrother et al., 2023; 2024; Grill et al., 2020; He et al., 2020). This protocol enables us to cleanly analyze the nature of the learned representations within the context of well-understood value function learning algorithms such as LSPE. In Appendix A, we include the pseudo-code for LSPE.

2.3 Stable, Realizable, and Generalizable Representations

We define stability of LSPE and related TD-methods following Ghosh & Bellemare (2020):

Definition 1 (Stability)**.** LSPE *is said to be stable if for any initial* $\theta_0 \in \mathbb{R}^d$, $\lim_{t\to\infty}\theta_t = \theta_{\text{LSPE}}$ *when* θ_t *is updated according to Equation (1).*

When determining the stability of LSPE, we have following proposition from prior work:

Proposition 1 (Asadi et al. (2024); Wang et al. (2021a))**.** LSPE *is stable if and only if the spectral radius of* $(\mathbb{E}[\Phi^\top\Phi])^{-1}(\gamma\mathbb{E}[\Phi^\top P^{\pi_e}\Phi])$, *i.e., its maximum absolute eigenvalue, is less than* 1.

Therefore, the stability of LSPE largely depends on the representations Φ and the distribution shift between the data distribution of $\mathcal{D}$ and π_e. In this work, we study the stability of LSPE for a fixed distribution of $\mathcal{D}$ and learn Φ. If a given Φ stabilizes LSPE, we say Φ is a *stable representation*.

In addition to stability, we also care about the realizability and generalizability of Φ. We say Φ is a *realizable* representation if $q^{\pi_e} \in \text{Span}(\Phi)$, where $\text{Span}(\Phi)$ is the subspace of all expressible action-value functions with Φ. Note that even if Φ is a realizable and stable representation, LSPE may not recover the q^{π_e} solution (Sutton & Barto, 2018). While generalization can have multiple interpretations, we say Φ *generalizes* well if the state-action features that are close in the representation space also have similar q^{π_e} values (Lyle et al., 2022; Lan et al., 2021).

2.4 Related Works

In this section, we discuss the most relevant prior literature on OPE and representation learning.

Representations for Offline RL and OPE. There are several works that have shown shaping representations can be effective for offline RL (Yang & Nachum, 2021; Islam et al., 2023; Nachum & Yang, 2024; Zang et al., 2023a; Arora et al., 2020; Uehara et al., 2021; Chen & Jiang, 2019; Pavse & Hanna, 2023b). Ghosh & Bellemare (2020) presented a theoretical understanding of how various representations can stabilize TD learning. However, they did not discuss bisimulation-based representations. Kumar et al. (2021); Ma et al. (2024); He et al. (2024) promote the stability of TD-based methods by increasing the rank of the representations to prevent representation collapse. However, as we show in Section 4, these types of representations can still lead to inaccurate OPE. On the other hand, KROPE mitigates representation collapse and leads to accurate OPE. Chang et al. (2022) introduced BCRL to learn Bellman complete representations for stable OPE. While in theory, BC representations are desirable, we found that BCRL is sensitive to hyperparameter tuning. In contrast, we show that KROPE is more robust to hyperparameter tuning. Pavse & Hanna (2023a) showed that bisimulation-based representations mitigate the divergence of FQE; however, they did not provide an explanation for divergence mitigation. Our work provides theoretical insight into the stability properties of bisimulation-based algorithms.

Bisimulation-based Representation Learning. Recently, there has been lot of interest in π-bisimulation algorithms for better generalization (Ferns et al., 2004; 2011; Ferns & Precup, 2014; Castro, 2019; Zang et al., 2023b). These algorithms measure similarity between two state-action pairs based on immediate rewards received and the similarity of next state-action pairs visited by π. These algorithms first define a distance metric that captures this π-bisimilarity, and then use this metric to learn representations such that π-bisimular states have similar representations (Castro et al., 2022; Castro, 2019; Zhang et al., 2021; Castro et al., 2023; Chen & Pan, 2022; Kemertas & Jepson, 2022). Zhang et al. (2021); Castro (2019) introduced a π-bisimulation learning algorithm but assume that the transition dynamics are either deterministic or Gaussian. Gelada et al. (2019) introduced a method closely related to bisimulation methods but required a reconstruction loss to work in practice. Castro et al. (2022) introduced MICO which allows for stochastic transition dynamics and no reconstruction loss, but was difficult to theoretically analyze. To overcome this difficulty, Castro et al. (2023) took a kernel perspective of π-bisimulation methods, which made their algorithm amenable to theoretical analysis. To the best of our knowledge, no works have studied the stability properties of π-bisimulation algorithms. In our work, we address this gap in the literature. We first extend Castro et al. (2023)'s kernel-based formulation from states to state-actions, and then show that this formulation stabilizes offline value function learning. The proofs for KROPE's basic theoretical properties (Section 3.1) follow those by Castro et al. (2023). Our stability-related theoretical results (Sections 3.2 and 3.3) and empirical analysis (Section 4) are novel to this work.

3 KERNEL REPRESENTATIONS FOR OFFLINE POLICY EVALUATION

We now present our bisimulation-based representation learning algorithm, kernel representations for OPE (KROPE). We present the desired KROPE kernel, define the KROPE operator, present its theoretical properties, prove stability properties of KROPE representations, and present a practical learning algorithm to learn them. We defer the proofs to Appendix B.

3.1 KROPE KERNEL AND OPERATOR

Prior π-bisimulation works define similarity between states in terms of the immediate rewards received and similarity of next states under π (see Figure 1) (Castro, 2019). In this work, we follow Castro et al. (2023) and define a kernel $k^{\pi_e} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ that captures this notion of similarity under π_e, but for pairs of state-actions. We refer to k^{π_e} as the KROPE kernel.

$$k^{\pi_e}(s_1, a_1; s_2, a_2) = k_1(s_1, a_1; s_2, a_2) + \gamma k_2(k^{\pi_e})(P^{\pi_e}(\cdot|s_1, a_1), P^{\pi_e}(\cdot|s_2, a_2)]. \tag{2}$$

where $k_1(s_1, a_1; s_2, a_2) := 1 - \frac{|r(s_1,a_1)-r(s_2,a_2)|}{|r_{\max}-r_{\min}|}$ and $k_2(k^{\pi_e})(P^{\pi_e}(\cdot|s_1, a_1), P^{\pi_e}(\cdot|s_2, a_2)) := \mathbb{E}_{s_1',a_1'\sim P^{\pi_e}(\cdot|s_1,a_1),s_2',a_2'\sim P^{\pi_e}(\cdot|s_2,a_2)}[k^{\pi_e}(s_1', a_1'; s_2', a_2')]$. Here, k_1 measures short-term similarity based on rewards received and k_2 measures long-term similarity between probability distributions by measuring similarity between samples of the distributions according to k^{π_e} (Castro et al., 2023).

Given this definition of the KROPE kernel, we now present an operator that converges to k^{π_e}:

Definition 2 (KROPE operator)**.** *Given a target policy π_e, the KROPE operator $\mathcal{F}^{\pi_e} : \mathbb{R}^{\mathcal{X}\times\mathcal{X}} \to \mathbb{R}^{\mathcal{X}\times\mathcal{X}}$ is defined as follows: for each kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, $\forall(s_1, a_1; s_2, a_2) \in \mathcal{X} \times \mathcal{X}$,*

$$\mathcal{F}^{\pi_e}(k)(s_1, a_1; s_2, a_2) := \underbrace{k_1(s_1, a_1; s_2, a_2)}_{\text{short-term similarity}} + \gamma \underbrace{\mathbb{E}_{s_1',s_2'\sim P, a_1',a_2'\sim\pi_e}[k(s_1', a_1'; s_2', a_2')]}_{\text{long-term similarity}} \tag{3}$$

where $s_1' \sim P(s_1'|s_1, a_1), s_2' \sim P(s_2'|s_2, a_2), a_1' \sim \pi_e(\cdot|s_1'), a_2' \sim \pi_e(\cdot|s_2')$, and $k_1(s_1, a_1; s_2, a_2) := 1 - \frac{|r(s_1,a_1)-r(s_2,a_2)|}{|r_{max}-r_{min}|}$ is a positive semidefinite kernel.

Proposition 2, proved in Appendix B.1, tells us that for some initial kernel k, repeatedly applying $\mathcal{F}^{\pi_e}$ to it will result in convergence k^{π_e}. Ultimately, k^{π_e} outputs a high (or low) similarity measure for two state-action pairs if their action-values under π_e are similar (or dissimilar). This intuition is formalized in Lemma 3, which states that the absolute action-value difference between any two state-action pairs under π_e is upper-bounded by the distance function induced by k^{π_e} plus an additive constant. Since KROPE's contraction, metric space completeness, and fixed-point uniqueness properties are similar to Castro et al. (2023)'s kernel, we defer the details to Appendix B.1.

3.2 STABILITY OF KROPE REPRESENTATIONS

In the previous section, we defined the KROPE kernel. Ultimately, however, we are interested in *representations* that satisfy the relationship in Equation (2). We modify Equation (2) accordingly by giving k^{π_e} some functional form in terms of state-action representations. We do so with the dot product: $\langle u, v\rangle = u^\top v, \forall u, v \in \mathbb{R}^d$, i.e., $k^{\pi_e}(s_1, a_1; s_2, a_2) = \phi(s_1, a_1)^\top \phi(s_2, a_2)$. With this setup, we write Equation (2) in matrix notation and define the KROPE representations as follows:

Definition 3 (KROPE Representations)**.** *Consider state-action representations $\Phi \in \mathbb{R}^{|\mathcal{X}|\times d}$ that are embedded in $\mathbb{R}^d$ with $k^{\pi_e}(s_1, a_1; s_2, a_2) = \phi(s_1, a_1)^\top \phi(s_2, a_2)$. We say Φ is a KROPE representation if it satisfies the following:*

$$\mathbb{E}_{\mathcal{D}}[\Phi\Phi^\top] = \mathbb{E}_{\mathcal{D}}[K_1] + \gamma \mathbb{E}_{\mathcal{D},\pi_e}[P^{\pi_e}\Phi(P^{\pi_e}\Phi)^\top] \tag{4}$$

where each entry of $K_1 \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$ represents the short-term similarity, k_1, between every pair of state-actions, i.e., $K_1(s_1, a_1; s_2, a_2) := 1 - \frac{|r(s_1,a_1)-r(s_2,a_2)|}{|r_{max}-r_{min}|}$.

Given this definition, we present our novel result proving the stability of KROPE representations:

Theorem 1. *If Φ is a KROPE representation as defined in Definition 3, then the spectral radius of $(\mathbb{E}[\Phi^\top\Phi])^{-1}\mathbb{E}[\gamma\Phi^\top P^{\pi_e}\Phi]$ is less than 1. That is, Φ stabilizes LSPE.*

Theorem 1, proved in Appendix B.2, tells us that KROPE representations stabilize OPE with LSPE. Intuitively, they are stable since when $(\mathbb{E}[\Phi^\top\Phi]))^{-1}\mathbb{E}[\gamma\Phi^\top P^{\pi_e}\Phi]$'s spectral radius is less than 1, each update to θ_t in Equation (1) is non-expansive. That is, each update brings θ_t closer to θ_{LSPE}.

3.3 Connection to Bellman Completeness

In this section, we draw a novel connection between KROPE representations and Bellman completeness. We say a function class $\mathcal{F}$ is Bellman complete if it is complete under the Bellman operator: $\mathcal{T}^{\pi_e}f \subseteq \mathcal{F}, \forall f \in \mathcal{F}$. For instance, suppose $\mathcal{F}$ is the class of linear functions spanned by Φ, $\mathcal{F} := \{f \in \mathbb{R}^{\mathcal{X}} : f := \Phi w\}, w \in \mathbb{R}^d$. Then if $\mathcal{T}^{\pi_e}f, \forall f \in \mathcal{F}$ is also a linear function within the span of Φ, we say Φ is a Bellman complete representation. Bellman completeness is an alternative condition for stability and is typically assumed to ensure to data-efficient policy evaluation (Wang et al., 2021b; Szepesvári & Munos, 2005; Chang et al., 2022). We now present our second main result. It states that KROPE representations are Bellman complete:

Theorem 2. *Let $\phi : \mathcal{X} \to \mathcal{X}^\phi$ be the state-action abstraction induced by grouping state-actions $x, y \in \mathcal{X}$ such that if $d_{\text{KROPE}}(x, y) = 0$, then $\phi(x) = \phi(y), \forall x, y \in \mathcal{X}$. Then ϕ is Bellman complete if the abstract reward function $r^\phi : \mathcal{X}^\phi \rightarrowtail (-1, 1)$ is injective (i.e., distinct abstract rewards).*

Takeaway #1: Stability of Bisimulation-based Representations

KROPE representations induce non-expansive value function updates and are Bellman complete. They avoid divergence of offline value function learning.

3.4 KROPE Learning Algorithm

In this section, we present an algorithm that learns the KROPE representations from data. We include the pseudo-code of KROPE in Appendix A. The KROPE learning algorithm uses an encoder $\phi_\omega : \mathcal{S}\times\mathcal{A} \to \mathbb{R}^d$, which is parameterized by weights ω of a function approximator. It then parameterizes the kernel with the dot product, i.e, $\tilde{k}_\omega(s_1, a_1; s_2, a_2) := \phi_\omega(s_1, a_1)^\top\phi_\omega(s_2, a_2)$ (see Equation (4)). Finally, the algorithm then minimizes the following loss function, which is similar to how the value function is learned in deep RL (Mnih et al., 2015):

$$\mathcal{L}_{\text{KROPE}}(\omega) := \mathbb{E}_{\mathcal{D}}\left[\left(\underbrace{1 - \frac{|r(s_1, a_1) - r(s_2, a_2)|}{|r_{\max} - r_{\min}|} + \gamma\mathbb{E}_{\pi_e}[\tilde{k}_{\bar{\omega}}(s_1', a_1'; s_2', a_2')]}_{\text{target estimate}} - \underbrace{\tilde{k}_\omega(s_1, a_1; s_2, a_2)}_{\text{current estimate}}\right)^2\right], \tag{5}$$

where the state-action pairs are sampled from $\mathcal{D}$, and $\bar{\omega}$ are weights of the target network that are periodically copied from ω (Mnih et al., 2015). In this work, we use KROPE as an auxiliary task, which introduces only a learning rate as an additional hyperparameter. We note that this fixed-point optimization procedure is similar to how the action-value function is learned in other RL fixed-point algorithms such as fitted q-evaluation (FQE) (Le et al., 2019).

4 Empirical Results

In this section, we present our empirical study designed to answer the following questions.

1. Does KROPE lead to stable representations with good realization and generalization?
2. Do KROPE representations lead to stable MSVE and low MSVE?

4.1 Empirical Setup

In this section, we describe the main details of our empirical setup. For further details such as datasets, policies, hyperparameters, and evaluation protocol please refer to Appendix C.

Baselines Our primary representation learning baseline is fitted q-evaluation (FQE) (Le et al., 2019). FQE is the most fundamental deep RL OPE algorithm that learns representations of state-actions to predict the long-term performance of a policy. While FQE is typically used as an OPE algorithm, it can also be viewed as a value-predictive representation learning algorithm (Lehnert & Littman, 2020). More specifically, consider its loss function: $\mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[\left(r(s,a)+\gamma\mathbb{E}_{a'\sim\pi_e}[q_{\bar{\xi}}(s',a')]-q_{\xi}(s,a)\right)^2\right]$, where $q_{\xi}(s,a) := \phi_{\xi'}(s,a)^\top w$ and $\xi = \{\xi', w\}$. We view ξ as the neural network weights of an action-value neural network and w as the linear weights of the network applied on the output of the penultimate layer $\phi_{\xi'}(s,a)$ of the neural network. Then minimizing this loss function shapes the representations $\phi_{\xi'}(s,a)$ to predict the expected future discounted return. As noted in Section 2, we follow the linear evaluation protocol where $\phi_{\xi'}$ is shaped by different auxiliary tasks and is then used with LSPE for OPE since it helps us understand the properties of the representations within the context of a well-understood value function learning algorithm (Grill et al., 2020; Chang et al., 2022; Farebrother et al., 2024; Wang et al., 2021a). We provide the pseudocode of this setup in Appendix A. We also note that in Appendix C, we present results of performing OPE using FQE instead LSPE, and find that KROPE still reliably produces stable OPE estimates.

We consider the following four non-bisimulation auxiliary representation learning algorithms that are typically paired with FQE for stability: 1) KROPE (ours), 2) BCRL-EXP-NA, which simultaneously optimizes three objectives: immediate reward prediction, next-state self-prediction loss, and maximization of the exploratory nature of $\mathbb{E}[\Phi^\top\Phi]$ by improving its condition number (Chang et al., 2022), 3) Absolute DR3 regularizer (Kumar et al., 2021; Ma et al., 2024), which promotes stability by minimizing the feature co-adaptation between successive features, i.e., $\phi(s,a)^\top\phi(s',a')$, and 4) BEER regularizer (He et al., 2024), which is similar to DR3 but lower bounds $\phi(s,a)^\top\phi(s',a')$. In all cases, the penultimate layer features of FQE's action-value encoder $\phi_{\xi'}$ are fed into LSPE for OPE. Our experiments focus on analyzing the properties of $\phi_{\xi'}$. Note that since BCRL was not designed as an auxiliary task (Chang et al., 2022), we evaluate it as a non-auxiliary (NA) task algorithm. We provide additional details on the baselines in Appendix C.

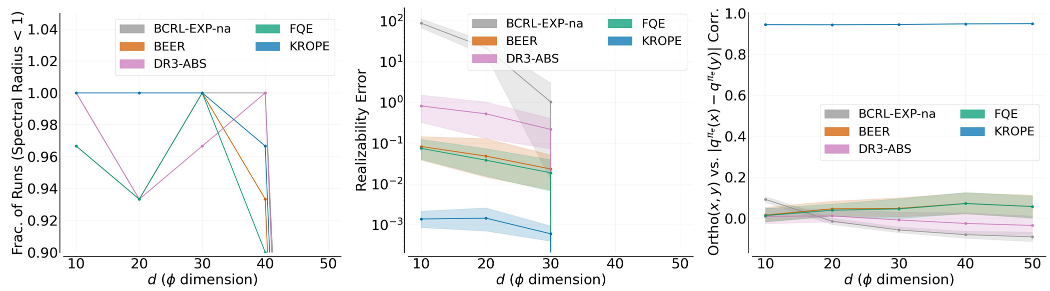
Domains We conduct our evaluation on a variety of domains: 1) Garnet MDPs, which are a class of tabular stochastic MDPs that are randomly generated given a fixed number of states and actions (Archibald et al., 1995); 2) 4 DM Control environments: CartPoleSwingUp, CheetahRun, FingerEasy, WalkerStand (Tassa et al., 2018); and 3) 9 D4RL datasets (Fu et al., 2020; 2021). The first domain enables us to analyze the algorithms' performance across a wide range of stochastic tabular MDPs. The second and third set of domains test the algorithms in continuous higher-dimensional state-action environments. Due to space constraints, we defer the D4RL results to Appendix C.3.2.

4.2 Analyzing Fundamental Properties of the Learned Representations

In this set of experiments on the Garnet MDPs domain, we answer our first question of whether KROPE representations lead to stable representations with good realization and generalization properties. We present the results in Figure 2. Our Garnet MDPs were generated with 8 states and 5 actions, with a total of $|\mathcal{X}| = 40$ state-actions, and each native (s,a) representation is a 1-hot vector. In these experiments, the native representation is fed into a linear encoder with a bias component and no activation function. All algorithms are trained for 500 epochs and we report the results by evaluating the final learned representations for different latent dimensions d.

Stability. Based on Theorem 1, a representation is stable if it induces a spectral radius of $(\mathbb{E}[\Phi^\top\Phi])^{-1}(\gamma\mathbb{E}[\Phi^\top P^{\pi_e}\Phi])$ that is less than 1. In Figure 2(a), we present the fraction of runs that result in such representations. We find that up till $d = 30$, 100% of KROPE and BEER runs have spectral radius less than 1. We also find that BCRL-EXP-NA produces stable representations up till $d = 40$. At $d = 50$, all algorithms produce unstable representations. These results suggest that KROPE, BEER, and BCRL-EXP-NA are reliable in producing stable representations when projecting state-actions into low dimensions. When $d \geq |\mathcal{X}|$, the covariance matrix $\mathbb{E}[\Phi^\top\Phi]$ is more likely to be a singular matrix, which implies higher chance of instability.

Realizability. A basic criterion for learning q^{π_e} is realizability. That is, we want $\epsilon := \|\Phi\hat{w} - q^{\pi_e}\|_2^2$, where $\hat{w} := \arg\min_w \|\Phi w - q^{\pi_e}\|_2^2$, to be low. In our experiments, we compute ϵ and plot it as a function of d in Figure 2(b). A critical message from our results is that stability and realizability do not always go hand-in-hand. While BCRL-EXP-NA has favorable spectral radius



(a) Spectral Radius (b) Realizability Error (c) Orthogonality

Figure 2: Evaluation of basic representation properties on Garnet MDPs with 40 state-actions vs. output dimension d. Figure 2(a): Fractions of runs out of 30 trials that resulted in spectral radius of $(\mathbb{E}[\Phi^\top \Phi])^{-1}(\gamma \mathbb{E}[\Phi^\top P^{\pi_e} \Phi])$ to be less than 1; higher is better. Figure 2(b): Realizability error; lower is better. Figure 2(c): Pearson correlation between orthogonality between pairs of latent features vs. their corresponding absolute q^{π_e} action-value difference; higher is better. All results are averaged over 30 trials and the shaded region is the 95% confidence interval.

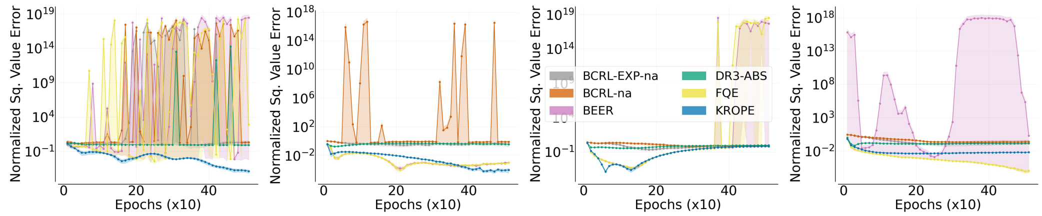
properties (Figure 2(a)), it has poor realizability, which will negatively affect its OPE accuracy. KROPE, on the other hand, has favorable stability and realizability properties up till $d = 30$. When $d \geq 40$, the realizability error is 0 for all algorithms since the subspace spanned by Φ is large enough to contain the true action-value function (Ghosh & Bellemare, 2020). While the realizability error is 0 for $d \geq 40$, the representations can be unstable (Figure 2(a)).

Generalization. Finally, we say that the representations have generalized well when state-actions that have similar q^{π_e} values are close to each other in the representation space (Lyle et al., 2022). We assess generalization by measuring the orthogonality: $1 - \frac{|\langle \phi(s_1,a_1), \phi(s_2,a_2) \rangle|}{\|\phi(s_1,a_1)\| \|\phi(s_2,a_2)\|}$ (Wang et al., 2024) between every state-action pair, $(s_1, a_1; s_2, a_2)$, and the absolute action-value difference: $|q^{\pi_e}(s_1, a_1) - q^{\pi_e}(s_2, a_2)|$. We then compute the Pearson correlation between these values for every pair and plot the correlation for each d in Figure 2(c). A correlation coefficient close to 1 indicates that the representations generalize well. We find that KROPE representations satisfy this property almost perfectly since it specifically tries to learn representations such that state-action pairs with similar values under π_e are similar. We observe that the other algorithms typically have zero or even negative correlation. A negative correlation indicates that state-actions with different action-values may be similar in latent space, which can result in higher realizability error (Figure 2(b)). A near-zero correlation but low realizability error such as in the case of FQE implies that accurate offline value prediction is still possible but that it generalizes poorly, which may slow down convergence to its OPE solution (Lyle et al., 2022).

4.3 Offline Policy Evaluation

In this set of experiments, we conduct experiments on four DM control environments to answer our second empirical question: whether KROPE representations lead to stable and low MSVE? We also evaluate BCRL-NA, which is BCRL without the exploration maximization regularizer. To stabilize training for all algorithms, we use wide neural networks with layernorm (Gallici et al., 2024; Ota et al., 2021). Note that while wide networks and layernorm stabilize training, they may not lead to stable LSPE under the linear evaluation protocol. During representation learning, we periodically evaluate the learned representations for OPE using LSPE. The corresponding (normalized) squared value errors are presented in Figure 3.

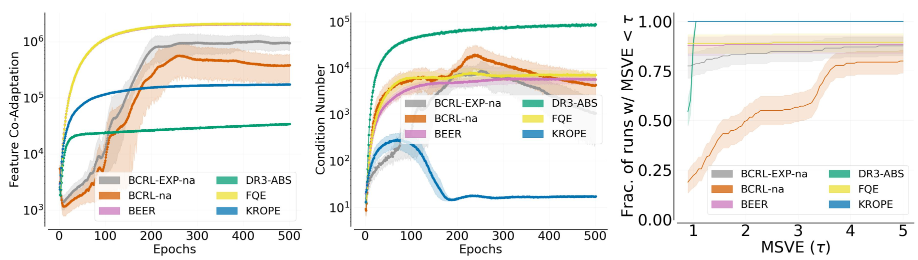
In general, we find that KROPE representations lead to low and stable MSVE. On the other hand, we find that the other auxiliary tasks inconsistently produce stable OPE estimates across all environments. For example, the performance of the ABS-DR3 and BEER regularizer suggests that explicitly trying to increase the rank of the features of the penultimate layer may hurt stability, and even if the OPE error is stable, it can hurt accuracy. We also make a similar observation for BCRL. However, in this case, we attribute poor performance to difficulty in optimizing the BCRL objective. In fact, in Figure 4(c), we will see that BCRL is sensitive to hyperparameter tuning. We also observe



(a) CartPoleSwingUp (b) CheetahRun (c) FingerEasy (d) WalkerStand

Figure 3: Normalized squared value error achieved by LSPE when using a particular representation vs. representation training epochs. LSPE estimates are computed every 10 epochs. Results are averaged over 20 trials and the shaded region is the 95% confidence interval. Lower and less erratic is better.

results consistent with a known result that BCRL-EXP-NA performs better than BCRL-NA indicating the known result that exploration maximization of the covariance matrix helps produce stable representations (Chang et al., 2022). Finally, while FQE achieves lower OPE error than KROPE on WalkerStand, it is very unstable on CartPoleSwingUp and Finger Easy, which motivates the need to shape the representations for stable and accurate OPE. We note that in the WalkerStand instance, the higher error of KROPE is unsurprising since Lemma 3 suggests that KROPE may lose realizability (see Appendix B.1). We refer the reader to Appendix C.1 for details of each algorithm. We also conduct the same experiment on 9 D4RL datasets and reach the similar conclusions (see Appendix C.3.2).



(a) Feature Co-adaptation (b) Condition Number (c) Hyperparameter Sensitivity

Figure 4: Stability-related metrics to understand the properties of KROPE on CartPoleSwingUp. Values are plotted as a function of training epochs. All results are averaged over 20 trials and shaded region is the 95% confidence interval. For hyperparameter sensitivity, larger area under the curve is better.

4.3.1 Analyzing Stability-Related Metrics

While KROPE performs well on the downstream task of OPE, it is important to analyze upstream stability-related metrics during the course of learning. These results give insight into the properties of the learned representations. We present the results in Figure 4 for CartPoleSwingUp and defer the remaining results to the Appendix C.

Feature co-adaptation. The feature-co-adaptation metric, i.e., $\sum_{(s,a,s')\in\mathcal{D},a'\sim\pi_e}\phi(s,a)^\top\phi(s',a')$ was shown to correlate with instability as high values can indicate representation collapse (Kumar et al., 2021). From Figure 4(a), we find that KROPE mitigates representation collapse of FQE (lowers feature co-adaptation from 10^6 to 10^5). Its relatively moderate value indicates that it is able to balance generalization and maintain distinctness between state-action representations. While low co-adaptation is preferred, extremely low values such as those achieved by ABS-DR3 do not necessarily lead to accurate OPE (see Figure 3). These results tell us which algorithms are more prone to representation collapse.

Condition number. Another metric that we analyze is the condition number of the covariance matrix $M := \mathbb{E}[\Phi^\top\Phi]$. A low condition number indicates numerical stability of the learning algorithm, i.e., solutions recovered by LSPE (see Equation (1)) or general TD algorithms (Asadi et al., 2024) are less likely to change with small perturbations to M (Chang et al., 2022). From Figure 4(b), we find that KROPE achieves lower condition number than FQE (from 10^4 to 10^1). As excepted, we observe that BCRL-EXP-NA achieves an even lower condition since it explicitly optimizes for a well-conditioned covariance matrix (10^4 to 10^3). While there is no ideal value, these results indicate the reliability of the representations in enabling TD-based algorithms to recover their value function solution. We observe qualitatively similar results for other environments (see Appendix C).

Hyperparameter Sensitivity. In OPE, hyperparameter tuning can be challenging since it may be infeasible to get access to ground truth performance of π_e (Fu et al., 2021). Therefore, we prefer algorithms that are robust to hyperparameter tuning, i.e, they reliably produce accurate OPE estimates for a wide range of hyperparameters. In Figure 4(c), we present the performance profile for each algorithm across *all* hyperparameter combinations and *all* trials (Agarwal et al., 2021). We tune the hyperparameters discussed in Appendix C.1. We find that 100% KROPE runs across all instances produce MSVE ≤ 1, which is not the case with other algorithms.

Takeaway #2: Practical Stable and Accurate Offline Policy Evaluation

OPE practitioners can use KROPE for stable and accurate evaluation of offline RL agents.

5 Limitations and Future Work

In this section, we discuss limitations and future work. A shortcoming of our work is that KROPE's learning algorithm is susceptible to instability since it is a semi-gradient method (Sutton & Barto, 2018). Moreover, its fixed-point optimization means it does not solve any objective function (Feng et al., 2019). In our work, we employed commonly-used techniques such as layernorm and wide neural networks to mitigate instability (Ota et al., 2021; Gallici et al., 2024). While these techniques potentially side-step the issue, the consequences of a semi-gradient method may still exist. In Appendix C.3.1, we present an empirical analysis to gauge when KROPE's learning algorithm may be unstable. We find that while *individual* off-policy transitions can determine the instability of fixed-point and semi-gradient algorithms such as FQE, *pairs* of off-policy transitions can determine KROPE's instability. Since we are unlikely to have control over the distribution over pairs of transitions in practice, we need to resort to fundamental changes to the algorithm. One potential change is based on that by Feng et al. (2019). Their key insight is to leverage the Legendre-Fenchel transformation from optimization theory and replace the fixed-point loss function of semi-gradient methods with an equivalent expression that avoids semi-gradient learning (Rockafellar & Wets, 1998). However, a drawback with this approach is that the new learning objective is a minimax procedure, which can be challenging to optimize in practice. In future work, we will explore the viability of this approach to design a provably convergent version of KROPE.

6 Conclusion

In this work, we tackled the problem of stabilizing offline value function learning in reinforcement learning. We introduced a bisimulation-based representation learning algorithm, kernel representations for OPE (KROPE), that shapes the state-action representations to stabilize this procedure. Theoretically, we showed that KROPE representations are stable from two perspectives: 1) non-expansiveness, i.e., they lead to value function learning updates that enable convergence to a fixed-point and 2) Bellman completeness, i.e., they satisfy a condition for data-efficient policy evaluation. Empirically, we showed that KROPE leads to more stable and accurate offline value function learning than baselines. Our work showed that bisimulation-based representation learning effectively stabilizes long-term performance evaluations of offline reinforcement learning agents.

References

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in Neural Information Processing Systems*, 34, 2021.

Thomas Welsh Archibald, K. I. M. McKinnon, and Lyn C. Thomas. On the generation of markov decision processes. *Journal of the Operational Research Society*, 46:354–361, 1995. URL https://api.semanticscholar.org/CorpusID:56653344.

Sanjeev Arora, Simon S. Du, Sham Kakade, Yuping Luo, and Nikunj Saunshi. Provable representation learning for imitation learning via bi-level optimization, 2020. URL https://arxiv.org/abs/2002.10544.

Kavosh Asadi, Shoham Sabach, Yao Liu, Omer Gottesman, and Rasool Fakoor. Td convergence: an optimization perspective. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2024. Curran Associates Inc.

Leemon C. Baird. Residual algorithms: Reinforcement learning with function approximation. In *International Conference on Machine Learning*, 1995. URL https://api.semanticscholar.org/CorpusID:621595.

Pablo Samuel Castro. Scalable methods for computing state similarity in deterministic Markov Decision Processes, November 2019. URL http://arxiv.org/abs/1911.09291. arXiv:1911.09291 [cs, stat].

Pablo Samuel Castro, Tyler Kastner, Prakash Panangaden, and Mark Rowland. MICo: Improved representations via sampling-based state similarity for Markov decision processes. *arXiv:2106.08229 [cs]*, January 2022. URL http://arxiv.org/abs/2106.08229. arXiv: 2106.08229.

Pablo Samuel Castro, Tyler Kastner, Prakash Panangaden, and Mark Rowland. A kernel perspective on behavioural metrics for markov decision processes. *TMLR*, 2023. URL https://openreview.net/forum?id=nHfPXl1ly7&referrer=%5BAuthor%20Console%5D(%2Fgroup%3Fid%3DTMLR%2FAuthors%23your-submissions).

Jonathan Chang, Kaiwen Wang, Nathan Kallus, and Wen Sun. Learning Bellman Complete Representations for Offline Policy Evaluation. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 2938–2971. PMLR, June 2022. URL https://proceedings.mlr.press/v162/chang22b.html. ISSN: 2640-3498.

Jianda Chen and Sinno Pan. Learning representations via a robust behavioral metric for deep reinforcement learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 36654–36666. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/eda9523faa5e7191aee1c2eaff669716-Paper-Conference.pdf.

Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning*, 2019. URL https://api.semanticscholar.org/CorpusID:141460093.

Jesse Farebrother, Joshua Greaves, Rishabh Agarwal, Charline Le Lan, Ross Goroshin, Pablo Samuel Castro, and Marc G Bellemare. Proto-value networks: Scaling representation learning with auxiliary tasks. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=oGDKSt9JrZi.

Jesse Farebrother, Jordi Orbay, Quan Vuong, Adrien Ali Taiga, Yevgen Chebotar, Ted Xiao, Alex Irpan, Sergey Levine, Pablo Samuel Castro, Aleksandra Faust, Aviral Kumar, and Rishabh Agarwal. Stop regressing: Training value functions via classification for scalable deep RL. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=dVpFKfqF3R.

Yihao Feng, Lihong Li, and Qiang Liu. A kernel loss for solving the bellman equation. *ArXiv*, abs/1905.10506, 2019. URL https://api.semanticscholar.org/CorpusID:166228108.

Norm Ferns and Doina Precup. Bisimulation metrics are optimal value functions. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI'14, pp. 210–219, Arlington, Virginia, USA, 2014. AUAI Press. ISBN 9780974903910.

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, UAI '04, pp. 162–169, Arlington, Virginia, USA, 2004. AUAI Press. ISBN 0974903906.

Norm Ferns, Prakash Panangaden, and Doina Precup. Bisimulation metrics for continuous markov decision processes. *SIAM Journal on Computing*, 40(6):1662–1714, 2011. doi: 10.1137/10080484X. URL https://doi.org/10.1137/10080484X.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning, 2020.

Justin Fu, Mohammad Norouzi, Ofir Nachum, George Tucker, Ziyu Wang, Alexander Novikov, Mengjiao Yang, Michael R. Zhang, Yutian Chen, Aviral Kumar, Cosmin Paduraru, Sergey Levine, and Thomas Paine. Benchmarks for deep off-policy evaluation. In *ICLR*, 2021. URL https://openreview.net/forum?id=kWSeGEeHvF8.

Scott Fujimoto, David Meger, Doina Precup, Ofir Nachum, and Shixiang Shane Gu. Why should i trust you, bellman? evaluating the bellman objective with off-policy data, 2022. URL https://openreview.net/forum?id=MUpxS9vDbZr.

Matteo Gallici, Mattie Fellows, Benjamin Ellis, Bartomeu Pou, Ivan Masmitja, Jakob Nicolaus Foerster, and Mario Martin. Simplifying deep temporal difference learning, 2024. URL https://arxiv.org/abs/2407.04811.

Carles Gelada, Saurabh Kumar, Jacob Buckman, Ofir Nachum, and Marc G. Bellemare. DeepMDP: Learning Continuous Latent Space Models for Representation Learning. Technical Report arXiv:1906.02736, arXiv, June 2019. URL http://arxiv.org/abs/1906.02736. arXiv:1906.02736 [cs, stat] type: article.

Dibya Ghosh and Marc G. Bellemare. Representations for stable off-policy reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL http://jmlr.org/papers/v13/gretton12a.html.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent a new approach to self-supervised learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

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 Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1861–1870. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/haarnoja18b.html.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9726–9735, 2020. doi: 10.1109/CVPR42600.2020.00975.

Qiang He, Tianyi Zhou, Meng Fang, and Setareh Maghsudi. Adaptive regularization of representation rank as an implicit constraint of bellman equation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=apXtolxDaJ.

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, 2023. URL https://arxiv.org/abs/2211.00164.

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 (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 4764–4777, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/256bf8e6923a52fda8ddf7dc050a1148-Abstract.html.

Mete Kemertas and Allan Jepson. Approximate policy iteration with bisimulation metrics, 2022. URL https://arxiv.org/abs/2202.02881.

Aviral Kumar, Rishabh Agarwal, Tengyu Ma, Aaron Courville, George Tucker, and Sergey Levine. Dr3: Value-based deep reinforcement learning requires explicit regularization, 2021. URL https://arxiv.org/abs/2112.04716.

Charline Le Lan, Marc G. Bellemare, and Pablo Samuel Castro. Metrics and continuity in reinforcement learning. 2021.

Hoang M. Le, Cameron Voloshin, and Yisong Yue. Batch Policy Learning under Constraints. In *International Conference on Machine Learning (ICML)*. arXiv, March 2019. URL http://arxiv.org/abs/1903.08738. arXiv:1903.08738 [cs, math, stat].

Lucas Lehnert and Michael L. Littman. Successor features combine elements of model-free and model-based reinforcement learning. *Journal of Machine Learning Research*, 21(196):1–53, 2020. URL http://jmlr.org/papers/v21/19-060.html.

Lihong Li, Thomas J Walsh, and Michael L Littman. Towards a Unified Theory of State Abstraction for MDPs. pp. 10, 2006.

Clare Lyle, Mark Rowland, Will Dabney, Marta Z. Kwiatkowska, and Yarin Gal. Learning dynamics and generalization in reinforcement learning. *ArXiv*, abs/2206.02126, 2022. URL https://api.semanticscholar.org/CorpusID:249146731.

Yi Ma, Hongyao Tang, Dong Li, and Zhaopeng Meng. Reining generalization in offline reinforcement learning via representation distinction. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2024. Curran Associates Inc.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. 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. *Nature*, 518(7540):529–533, February 2015. ISSN 00280836. URL http://dx.doi.org/10.1038/nature14236.

Ofir Nachum and Mengjiao Yang. Provable representation learning for imitation with contrastive fourier features. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9781713845393.

Angelia Nedic and Dimitri Bertsekas. Least squares policy evaluation algorithms with linear function approximation. *Discrete Event Dynamic Systems*, 13:79–110, 01 2003. doi: 10.1023/A:1022192903948.

Keita Ota, Devesh K. Jha, and Asako Kanezaki. Training larger networks for deep reinforcement learning. *ArXiv*, abs/2102.07920, 2021. URL https://api.semanticscholar.org/CorpusID:231934052.

Vern I. Paulsen and Mrinal Raghupathi. *An Introduction to the Theory of Reproducing Kernel Hilbert Spaces*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2016.

Brahma S. Pavse and Josiah P. Hanna. State-Action Similarity-Based Representations for Off-Policy Evaluation. In *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, December 2023a.

Brahma S Pavse and Josiah P Hanna. Scaling Marginalized Importance Sampling to High-Dimensional State-Spaces via State Abstraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, February 2023b.

Doina Precup, Richard S. Sutton, and Satinder P. Singh. Eligibility traces for off-policy policy evaluation. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, pp. 759–766, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1558607072.

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

R. Tyrrell Rockafellar and Roger J.-B. Wets. *Variational Analysis*. Springer Verlag, Heidelberg, Berlin, New York, 1998.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html.

Csaba Szepesvari. *Algorithms for Reinforcement Learning*. Morgan and Claypool Publishers, 2010. ISBN 1608454924.

Csaba Szepesvári and Rémi Munos. Finite time bounds for sampling based fitted value iteration. In *Proceedings of the 22nd International Conference on Machine Learning*, ICML '05, pp. 880–887, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1595931805. doi: 10.1145/1102351.1102462. URL https://doi.org/10.1145/1102351.1102462.

Yunhao Tang, Zhaohan Daniel Guo, Pierre Harvey Richemond, Bernardo Ávila Pires, Yash Chandak, Rémi Munos, Mark Rowland, Mohammad Gheshlaghi Azar, Charline Le Lan, Clare Lyle, András György, Shantanu Thakoor, Will Dabney, Bilal Piot, Daniele Calandriello, and Michal Valko. Understanding self-predictive learning for reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy P. Lillicrap, and Martin A. Riedmiller. Deepmind control suite. *CoRR*, abs/1801.00690, 2018. URL http://arxiv.org/abs/1801.00690.

J.N. Tsitsiklis and B. Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Transactions on Automatic Control*, 42(5):674–690, 1997. doi: 10.1109/9.580874.

Masatoshi Uehara, Xuezhou Zhang, and Wen Sun. Representation learning for online and offline rl in low-rank mdps. *arXiv preprint arXiv:2110.04652*, 2021.

Cameron Voloshin, Hoang Minh Le, Nan Jiang, and Yisong Yue. Empirical study of off-policy policy evaluation for reinforcement learning. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021. URL https://openreview.net/forum?id=IsK8iKbL-I.

Han Wang, Erfan Miahi, Martha White, Marlos C. Machado, Zaheer Abbas, Raksha Kumaraswamy, Vincent Liu, and Adam White. Investigating the properties of neural network representations in reinforcement learning. *Artificial Intelligence*, 330:104100, 2024. ISSN 0004-3702. doi: https://doi.org/10.1016/j.artint.2024.104100. URL https://www.sciencedirect.com/science/article/pii/S0004370224000365.

Ruosong Wang, Yifan Wu, Ruslan Salakhutdinov, and Sham Kakade. Instabilities of Offline RL with Pre-Trained Neural Representation. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 10948–10960. PMLR, July 2021a. URL https://proceedings.mlr.press/v139/wang21z.html. ISSN: 2640-3498.

Ruosong Wang, Yifan Wu, Ruslan Salakhutdinov, and Sham M. Kakade. Instabilities of offline RL with pre-trained neural representation. *CoRR*, abs/2103.04947, 2021b. URL https://arxiv.org/abs/2103.04947.

Mengjiao Yang and Ofir Nachum. Representation Matters: Offline Pretraining for Sequential Decision Making. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 11784–11794. PMLR, July 2021. URL https://proceedings.mlr.press/v139/yang21h.html. ISSN: 2640-3498.

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, 2023a. URL https://arxiv.org/abs/2211.00863.

Hongyu Zang, Xin Li, Leiji Zhang, Yang Liu, Baigui Sun, Riashat Islam, Remi Tachet des Combes, and Romain Laroche. Understanding and addressing the pitfalls of bisimulation-based representations in offline reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b. URL https://openreview.net/forum?id=sQyRQjun46.

Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning Invariant Representations for Reinforcement Learning without Reconstruction, April 2021. URL http://arxiv.org/abs/2006.10742. arXiv:2006.10742 [cs, stat].

A BACKGROUND

In this section, we present the theoretical background.

A.1 BISIMULATION METRICS

In this section, we present background information on bisimulations and its associated metrics. Our proposed representation learning algorithm is a bisimulation-based algorithm. Bisimulation abstractions are those under which two states with identical reward functions and that lead to identical groups of next states under any action are classified as similar (Ferns et al., 2004; 2011; Ferns & Precup, 2014). Bisimulations are the strictest forms of abstractions. In practice, the exact bisimulation criterion is difficult to satisfy computationally and statistically. A more relaxed version of this notion is the π-*bisimulation metrics*. These metrics capture the similarity between two states such that two states are considered similar if they have identical *expected* reward functions and *expected* transitions to identical groups of next states under a *fixed policy* π (Castro, 2019).

We first give the definition of bisimulation.

Definition 4. *(Li et al., 2006) An abstraction $\phi : \mathcal{S} \to \mathcal{S}^\phi$ over the state space $\mathcal{S}$ is a bisimulation if for any action a and any abstract state $s^\phi \in \mathcal{S}^\phi$, ϕ is such that for any two states $s_1, s_2 \in \mathcal{X}$, $\phi(s_1) = \phi(s_2)$ implies that $r(s_1, a) = r(s_2, a)$ and $\sum_{s' \in s^\phi} P(s'|s_1, a) = \sum_{s' \in s^\phi} P(s'|s_2, a)$.*

Below we define π-bisimulations for state-actions instead of states:

Definition 5. *(Castro, 2019) An abstraction $\phi : \mathcal{X} \to \mathcal{X}^\phi$ over the state-action space $\mathcal{X}$ is a π-bisimulation for a fixed policy π if for any two state-actions $x, y \in \mathcal{X}$ and abstract state-action $x^\phi \in \mathcal{X}^\phi$, ϕ is such that $\phi(x) = \phi(y)$ implies that $r(x) = r(y)$ and $\sum_{x' \in x^\phi} P^\pi(x'|x) = \sum_{x' \in x^\phi} P^\pi(x'|y)$.*

The above definitions are based on exact groupings between state-actions. This strictness motivates the use of bisimulation and π-bisimulation metrics, which we define below.

Theorem 3. *(Ferns et al., 2004) Let $\mathcal{M}(\mathcal{S})$ be the space of bounded pseudometrics on the state-space $\mathcal{S}$. Then define $\mathcal{B} : \mathcal{M}(\mathcal{S}) \to \mathcal{M}(\mathcal{S})$ such that for each $d \in \mathcal{M}(\mathcal{S})$:*

$$\mathcal{B}(d)(s_1, s_2) = \max_{a \in \mathcal{A}}(|r(s_1, a) - r(s_2, a)| + \gamma \mathcal{W}(d)(P(\cdot|s_1, a), P(\cdot|s_2, a)),$$

where $\mathcal{W}$ is Wasserstein distance between the two distributions under metric d. Then $\mathcal{B}$ has a unique fixed point, d^*, and d^* is a bisimulation metric.

Similarly, we have the π-bisimulation metric:

Theorem 4. *(Castro, 2019) Let $\mathcal{M}(\mathcal{X})$ be the space of bounded pseudometrics on the state-action space $\mathcal{X}$ and π be a fixed policy. Then define $\mathcal{B} : \mathcal{M}(\mathcal{X}) \to \mathcal{M}(\mathcal{X})$ such that for each $d \in \mathcal{M}(\mathcal{S})$:*

$$\mathcal{B}(d)(x, y) = |r(x) - r(y)| + \gamma \mathcal{W}(d)(P^\pi(\cdot|x), P^\pi(\cdot|y)),$$

where $\mathcal{W}$ is Wasserstein distance between the two distributions under metric d. Then $\mathcal{B}$ has a unique fixed point, d^*, and d^* is a π-bisimulation metric.

Using the above metrics, prior works have introduced several representation learning algorithms to learn representations such that the distance between representations in latent space model the above distance metrics (Castro et al., 2022; 2023; Zhang et al., 2021; Kemertas & Aumentado-Armstrong, 2021; Pavse & Hanna, 2023a).

A.2 REPRODUCING KERNEL HILBERT SPACES

Let $\mathcal{X}$ be a finite set and define a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ to be a positive semidefinite kernel if it is symmetric and positive semidefinite. We then have for any $\{x_1, x_2, ..., x_n\} \in \mathcal{X}$ and $\{c_1, c_2, ..., c_n\} \in \mathbb{R}$:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} c_i, c_j k(x_i, x_j) \geq 0$$

Note that if the above inequality is strictly greater than zero whenever $\{c_1, \ldots, c_n\}$ has at least one nonzero, we say the kernel is positive definite. Given a kernel k on $\mathcal{X}$ with the reproducing property, we can construct a space of functions $\mathcal{H}_k$ referred to as a reproducing kernel Hilbert space (RKHS) with the following steps:

1. Construct a vector space of real-valued functions on $\mathcal{X}$ of the form $\{k(x, \cdot) : x \in \mathcal{X}\}$.
2. Equip this space with an inner product given by $\langle k(x, \cdot), k(y, \cdot)\rangle_{\mathcal{H}_k} = k(x, y)$.
3. Take the completion of the vector space with respect to the above inner product.

Our resulting vector space $\mathcal{H}_k$ is then an RKHS.

It is often convenient to write $\psi(x) := k(x, \cdot) \in \mathcal{H}_k$, which is called the feature map and is an embedding of x in $\mathcal{H}_k$. One can also embed probability distributions into $\mathcal{H}_k$. That is, $\Phi : \mathcal{P}(\mathcal{X}) \to \mathcal{H}_k$, which maps probability distributions over $\mathcal{X}$ to $\mathcal{H}_k$. We define $\Phi(\mu) = \mathbb{E}_{X\sim\mu}[\psi(X)]$, which is the mean embedding in $\mathcal{H}_k$ under μ.

Given these embeddings in the Hilbert space, we can quantify the distances between elements in $\mathcal{X}$ and $\mathcal{P}(\mathcal{X})$ in terms of the embeddings.

Definition 6. *Given a positive semidefinite kernel k, define ρ_k as its induced distance:*

$$\rho_k := \|\psi(x) - \psi(y)\|_{\mathcal{H}_k}.$$

By expanding the inner product, the squared distance can be written in terms of k:

$$\rho_k^2(x, y) = k(x, x) + k(y, y) - 2k(x, y).$$

Similarly, we have distances on $\mathcal{P}(\mathcal{X})$ using Φ:

Definition 7. *(Gretton et al., 2012) Let k be a kernel on $\mathcal{X}$ and $\Phi : \mathcal{P}(\mathcal{X}) \to \mathcal{H}_k$ be as defined above. Then the Maximum Mean Discrepancy (MMD) is a pseudo metric on $\mathcal{P}(\mathcal{X})$ defined by:*

$$\mathit{MMD}(k)(\mu, \nu) = \|\Phi(\mu) - \Phi(\nu)\|_{\mathcal{H}_k}.$$

The core usage of the RKHS is to precisely characterize the nature of the KROPE kernel. In practice, we deal with neural network representations, which are embedded in Euclidean space. Therefore, our goal is to learn representations in Euclidean space that approximate the properties of representations in the RKHS. For more details on the RKHS, we refer readers to Castro et al. (2023) and Gretton et al. (2012).

A.3 Algorithm Pseudocode

In this section, we present the pseudocode for LSPE and for our FQE + auxiliary task with LSPE for OPE setup.

Algorithm 1 LSPE

1: Input: policy to evaluate π_e, batch $\mathcal{D}$, fixed encoder function $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$.
2: Initialize $\theta_0 \in \mathbb{R}^d$ randomly.
3: Apply ϕ to $\mathcal{D}$ to generate Φ.
4: **for** t = 0, 1, 2, ... $T - 1$ **do**
5: $\quad\theta_{t+1} \leftarrow (\mathbb{E}[\Phi^\top\Phi])^{-1}\mathbb{E}[\Phi^\top(r + \gamma P^{\pi_e}\Phi\theta_t)]$
6: **end for**
7: Return θ_T

Algorithm 2 FQE + representation learning auxiliary task with LSPE for OPE

1: Input: policy to evaluate π_e, batch $\mathcal{D}$, encoder parameters class Ω, encoder function $\phi : \mathcal{S}\times\mathcal{A} \to \mathbb{R}^d$, action-value linear function $q : \mathbb{R}^d \to \mathbb{R}$, $\alpha \in [0, 1]$.
2: **for** epoch = 1, 2, 3, ... T **do**
3: $\mathcal{L}(\omega) := \alpha\text{Aux-Task}(\phi_\omega, \mathcal{D}, \pi_e)+(1-\alpha)\mathbb{E}_{(s,a,s')\sim\mathcal{D}}\left[\left(r(s,a) + \gamma\mathbb{E}_{a'\sim\pi_e}[q_{\bar{\xi}}(\phi_{\hat{\omega}}(s',a'))] - q_\xi(\phi_{\hat{\omega}}(s,a))\right)^2\right]$ {where the penultimate features ϕ are fed into an auxiliary representation learning algorithm such as KROPE, DR3, BEER etc.}
4: $\hat{\omega}_t := \arg\min_{\omega\in\Omega} \mathcal{L}(\omega)$
5: Periodically run LSPE, $\theta :=$ LSPE$(\pi_e, \mathcal{D}, \phi_\omega)$.
6: Compute estimated action-values, $\hat{q} := \Phi_{\hat{\omega}_t}\theta$. {where $\phi_{\hat{\omega}}$ is applied to $\mathcal{D}$ to get Φ_ω}
7: **end for**
8: Return $\hat{q} := \Phi_{\hat{\omega}_T}\theta$. {Estimated action-value function of π_e, q^{π_e}.}

B Theoretical Results

In this section, we present the proofs of our main and supporting theoretical results. The first set of proofs in Section B.1 show that KROPE is a valid operator. While new to our work, the proofs follow those by Castro et al. (2023). The next set of proofs in Section B.2 prove the stability of KROPE representations and are novel to our work. For presentation purposes, it will often be convenient to refer to a state-action pair as $x \in \mathcal{X}$ instead of (s, a).

B.1 KROPE Operator Validity

We now present the proofs demonstrating the validity of the KROPE operator. All the proofs in this sub-section model those by Castro et al. (2023). The primary difference is that our operator is for state-actions instead of states.

Lemma 1. *Let $\mathcal{K}(\mathcal{X})$ be the space of positive semidefinite kernels on $\mathcal{X}$. The* KROPE *operator $\mathcal{F}^{\pi_e}$ is a contraction with modulus γ in $\|\cdot\|_\infty$.*

Proof. Let $k_1, k_2 \in \mathcal{K}(\mathcal{X})$. We then have:

$$\begin{aligned}
&\|\mathcal{F}^{\pi_e}(k_1) - \mathcal{F}^{\pi_e}(k_2)\|_\infty \\
&= \max_{(x,y)\in\mathcal{X}\times\mathcal{X}} |\mathcal{F}^{\pi_e}(k_1)(x,y) - \mathcal{F}^{\pi_e}(k_2)(x,y)| \\
&= \gamma \max_{(x,y)\in\mathcal{X}\times\mathcal{X}} |\mathbb{E}_{X'\sim P^{\pi_e}(\cdot|x),Y'\sim P^{\pi_e}(\cdot|y)}[k_1(X',Y')] - \mathbb{E}_{X'\sim P^{\pi_e}(\cdot|x),Y'\sim P^{\pi_e}(\cdot|y)}[k_2(X',Y')]| \\
&= \gamma \max_{(x,y)\in\mathcal{X}\times\mathcal{X}} |\mathbb{E}_{X'\sim P^{\pi_e}(\cdot|x),Y'\sim P^{\pi_e}(\cdot|y)}[k_1(X',Y') - k_2(X',Y')]| \\
&\le \gamma\|k_1 - k_2\|_\infty.
\end{aligned}$$

This completes the proof of the lemma. □

Lemma 2. *Let $\mathcal{K}(\mathcal{X})$ be the space of positive semidefinite kernels on $\mathcal{X}$. Then the metric space $(\mathcal{K}(\mathcal{X}), \|\cdot\|_\infty)$ is complete.*

Proof. To show that $\mathcal{K}(\mathcal{X})$ is complete it suffices to show that every Cauchy sequence $\{k_n\}_{n\ge0}$ has a limiting point in $\mathcal{K}(\mathcal{X})$. Since $\mathcal{X}$ is a finite, the space of function $\mathbb{R}^{\mathcal{X}\times\mathcal{X}}$ is a finite-dimensional vector space, which is complete under $\|\cdot\|_\infty$. Thus, the limiting point $k \in \mathbb{R}^{\mathcal{X}\times\mathcal{X}}$ of the Cauchy sequence $\{k_n\}_{n\ge0}$ lies in $\mathbb{R}^{\mathcal{X}\times\mathcal{X}}$. Moreover, since we are considering only positive semidefinite kernel elements in the Cauchy sequence and they uniformly converge to $k \in \mathbb{R}^{\mathcal{X}\times\mathcal{X}}$, k must also be positive semidefinite. Thus, $\mathcal{K}(\mathcal{X})$ is complete under $\|\cdot\|_\infty$. □

Proposition 2. *The* KROPE *operator $\mathcal{F}^{\pi_e}$ has a unique fixed point in $\mathcal{K}(\mathcal{X})$. That is, there is a unique kernel $k^{\pi_e} \in \mathcal{K}(\mathcal{X})$ satisfying*

$$k^{\pi_e}(s_1, a_1; s_2, a_2) = 1 - \frac{|r(s_1,a_1) - r(s_2,a_2)|}{|r_{max} - r_{min}|} + \gamma\mathbb{E}_{s_1',s_2'\sim P,a_1',a_2'\sim\pi_e}[k^{\pi_e}(s_1', a_1'; s_2', a_2')].$$

Proof. Due to Lemmas 1 and 2, $\mathcal{F}^{\pi_e}$ is a contraction in a complete metric space. Therefore, by Banach's fixed point theorem, the unique fixed point k^{π_e} exists. □

Proposition 3. *The* KROPE *similarity metric* d_{KROPE} *satisfies:*

$$\forall x, y \in \mathcal{X}, d_{\text{KROPE}}(x, y) = |r(x) - r(y)| + \gamma \mathit{MMD}^2(k^{\pi_e})(P^{\pi_e}(\cdot|x), P^{\pi_e}(\cdot|y)).$$

Proof. To see this fact, we can write out the squared Hilbert space distance:

$$\begin{aligned} d_{\text{KROPE}}(x, y) &= \|\psi^{\pi_e}(x) - \psi^{\pi_e}(y)\|^2_{\mathcal{H}_k^{\pi_e}} \\ &= k^{\pi_e}(x, x) + k^{\pi_e}(y, y) - 2k^{\pi_e}(x, y) \\ &= |r(x) - r(y)| + \gamma\langle\Phi(P^{\pi_e}(\cdot|x)), \Phi(P^{\pi_e}(\cdot|x))\rangle_{\mathcal{H}_k^{\pi_e}} + \gamma\langle\Phi(P^{\pi_e}(\cdot|y)), \Phi(P^{\pi_e}(\cdot|y))\rangle_{\mathcal{H}_k^{\pi_e}} \\ &\qquad - 2\gamma\langle\Phi(P^{\pi_e}(\cdot|x)), \Phi(P^{\pi_e}(\cdot|y))\rangle_{\mathcal{H}_k^{\pi_e}} \\ &= |r(x) - r(y)| + \gamma \text{MMD}^2(k^{\pi_e})(P^{\pi_e}(\cdot|x), P^{\pi_e}(\cdot|y)), \end{aligned}$$

where the third line uses

$$k^{\pi_e}(x, x) = \gamma \mathbb{E}_{X_1', X_2' \sim P^{\pi_e}(\cdot|x)}[k^{\pi_e}(X_1', X_2')] = \gamma\langle\Phi(P^{\pi_e}(\cdot|x), P^{\pi_e}(\cdot|x)\rangle_{\mathcal{H}_k^{\pi_e}}.$$

This completes the proof. □

Before presenting Lemma 3, we define the distance metric $d_{\text{KROPE}} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ induced by the KROPE kernel k^{π_e} as follows:

$$\forall x, y \in \mathcal{X} : \; d_{\text{KROPE}}(x, y) := k^{\pi_e}(x, x) + k^{\pi_e}(y, y) - 2k^{\pi_e}(x, y).$$

Lemma 3. *We have* $|q^{\pi_e}(x) - q^{\pi_e}(y)| \leq d_{\text{KROPE}}(x, y) + C$, *where* $C = \frac{1}{2}\sum_{n\geq 0} \gamma^n(\Delta_n^{\pi_e}(x) + \Delta_n^{\pi_e}(y))$ *and* $\Delta_n^{\pi_e}(x) = \mathbb{E}_{X' \sim (P^{\pi_e}(\cdot|x))^n}\left[\mathbb{E}_{X_1'', X_2'' \sim P^{\pi_e}(\cdot|X')}\left[|r(X_1'') - r(X_2'')|\right]\right]$.

Proof. We will prove this with induction. We first define the relevant terms involved. We consider the sequences of functions $\{k_m\}_{m\geq 0}$ and $\{q_m\}_{m\geq 0}$, where $k_0, q_0 = 0$. Since $\mathcal{F}^{\pi_e}$ and $\mathcal{T}^{\pi_e}$ are contraction mappings, we know that $\lim_{m\to\infty} k_m = k^{\pi_e}$ and $\lim_{m\to\infty} q_m = q^{\pi_e}$ as $\mathcal{F}^{\pi_e}$ and $\mathcal{T}^{\pi_e}$ are applied respectively at each iteration m. At the mth application of the operators, we have the corresponding kernel function k_m along with its induced distance function $d_m(x, y) = k_m(x, x) + k_m(y, y) - 2k_m(x, y)$. We will now prove the following for all m:

$$|q_m(x) - q_m(y)| \leq d_m(x, y) + \frac{1}{2}\sum_{n\geq 0}^{m} \gamma^n(\Delta_n^{\pi_e}(x) + \Delta_n^{\pi_e}(y)) \tag{6}$$

where $\Delta_n^{\pi_e}(x) = \mathbb{E}_{X' \sim (P^{\pi_e}(\cdot|x))^n}[\mathbb{E}_{X_1'', X_2'' \sim P^{\pi_e}(\cdot|X')}[|r(X_1'') - r(X_2'')|]]$.

The base case $m = 0$ follows immediately since the LHS is zero while the RHS can be non-zero. We now assume the induction hypothesis in Equation (6) is true. We then consider iteration $m + 1$:

$$\begin{aligned} &|q_{m+1}(x) - q_{m+1}(y)| \\ &= \left|r(x) + \gamma\mathbb{E}_{X' \sim P^{\pi_e}(\cdot|x)}[q_m(X')] - r(y) - \gamma\mathbb{E}_{Y' \sim P^{\pi_e}(\cdot|y)}[q_m(Y')]\right| \\ &\leq |r(x) - r(y)| + \gamma\mathbb{E}_{X' \sim P^{\pi_e}(\cdot|x), Y' \sim P^{\pi_e}(\cdot|y)}[|q_m(X') - q_m(Y')|] \\ &\leq |r(x) - r(y)| + \gamma\mathbb{E}_{X' \sim P^{\pi_e}(\cdot|x), Y' \sim P^{\pi_e}(\cdot|y)}\left[d_m(X', Y') + \frac{1}{2}\sum_{n=0}^{m}\gamma^n(\Delta_n^{\pi_e}(X') + \Delta_n^{\pi_e}(Y'))\right] \\ &= |r(x) - r(y)| + \gamma\mathbb{E}_{X' \sim P^{\pi_e}(\cdot|x), Y' \sim P^{\pi_e}(\cdot|y)}\left[d_m(X', Y') + \frac{1}{2}\sum_{n=1}^{m+1}\gamma^n(\Delta_n^{\pi_e}(x) + \Delta_n^{\pi_e}(y))\right] \end{aligned}$$

where we have used the fact that $\mathbb{E}_{X' \sim P^{\pi_e}(\cdot|x)}[\Delta_n^{\pi_e}(X')] = \Delta_{n+1}^{\pi_e}(x)$. We can then proceed from above as follows:

$$
\begin{aligned}
&= |r(x) - r(y)| + \gamma \mathbb{E}_{X' \sim P^{\pi_e}(\cdot|x), Y' \sim P^{\pi_e}(\cdot|y)} \left[d_m(X', Y') + \frac{1}{2} \sum_{n=1}^{m+1} \gamma^n (\Delta_n^{\pi_e}(x) + \Delta_n^{\pi_e}(y)) \right] \\
&\leq |r(x) - r(y)| + \gamma \mathbb{E}_{X' \sim P^{\pi_e}(\cdot|x), Y' \sim P^{\pi_e}(\cdot|y)} [d_m(X', Y')] \\
&\quad + \frac{1}{2} \mathbb{E}_{\substack{X_1', X_2' \sim P^{\pi_e}(\cdot|x) \\ Y_1', Y_2' \sim P^{\pi_e}(\cdot|y)}} [|r(X_1') - r(X_2')| + |r(Y_1') - r(Y_2')|] \\
&\quad + \frac{1}{2} \sum_{n=1}^{m+1} \gamma^n (\Delta_n^{\pi_e}(x) + \Delta_n^{\pi_e}(y)) \\
&= |r(x) - r(y)| + \gamma \mathbb{E}_{X' \sim P^{\pi_e}(\cdot|x), Y' \sim P^{\pi_e}(\cdot|y)} [d_m(X', Y')] + \frac{1}{2} \sum_{n=0}^{m+1} \gamma^n (\Delta_n^{\pi_e}(x) + \Delta_n^{\pi_e}(y)) \\
&= d_{m+1}(x, y) + \frac{1}{2} \sum_{n=0}^{m+1} \gamma^n (\Delta_n^{\pi_e}(x) + \Delta_n^{\pi_e}(y))
\end{aligned}
$$

We thus have $|q_{m+1}(x) - q_{m+1}(y)| \leq d_{m+1}(x, y) + \frac{1}{2} \sum_{n=0}^{m+1} \gamma^n (\Delta_n^{\pi_e}(x) + \Delta_n^{\pi_e}(y))$, which completes the proof. □

Lemma 3 tells us that the KROPE state-actions that are close in latent space also have similar action-values upto a constant $C := \frac{1}{2} \sum_{n=0}^{m+1} \gamma^n (\Delta_n^{\pi_e}(x) + \Delta_n^{\pi_e}(y))$. Intuitively, $\Delta_n^{\pi_e}(x)$ is the expected absolute reward difference between two trajectories at the nth step after π_e is rolled out from x. If the transition dynamics and π_e are deterministic, we have $C = 0$ (Castro, 2019; Zhang et al., 2021). Note that while the deterministic transition dynamics assumption is eliminated, the bound suggests that KROPE may hurt accuracy of $\hat{q}^{\pi_e}$ since when $d_{\text{KROPE}}(x, y) = 0$, we get $|q^{\pi_e}(x) - q^{\pi_e}(y)| \leq C$. This indicates that two state-actions that may have different action-values are considered the same under KROPE. This implies that while x and y should have different representations, they actually may have the same representation.

B.2 KROPE STABILITY

In this section we present our main results. We present supporting theoretical results in Section B.2.1 and main theoretical results in Section B.3. To the best of our knowledge, even the supporting proofs in Section B.2.1 are new.

B.2.1 SUPPORTING THEORETICAL RESULTS

We present the following definitions that we refer to in our proofs.

Definition 8 (Bellman completeness (Chen & Jiang, 2019))**.** *The function class $\mathcal{F}$ is said to be Bellman complete if $\forall f \in \mathcal{F}$, it holds that $\mathcal{T}^{\pi_e} f \in \mathcal{F}$. That is $\sup_{f \in \mathcal{F}} \inf_{g \in \mathcal{F}} \|g - \mathcal{T}^{\pi_e} f\|_\infty = 0$, where $\mathcal{F} \subset \mathcal{X} \to [\frac{r_{min}}{1-\gamma}, \frac{r_{max}}{1-\gamma}]$, and $\mathcal{T}^{\pi_e}$ is the Bellman operator.*

Definition 9 (Piece-wise constant functions (Chen & Jiang, 2019))**.** *Given a state-action abstraction ϕ, let $\mathcal{F}^\phi \subset \mathcal{X} \to [\frac{r_{min}}{1-\gamma}, \frac{r_{max}}{1-\gamma}]$. Then $f \in \mathcal{F}^\phi$ is said to be a piece-wise constant function if $\forall x, y \in \mathcal{X}$ where $\phi(x) = \phi(y)$, we have $f(x) = f(y)$.*

Proposition 4. *If a state-action abstraction function $\phi : \mathcal{X} \to \mathcal{X}^\phi$ is a π_e-bisimulation abstraction, then $\mathcal{F}^\phi$ is Bellman complete, that is, $\sup_{f \in \mathcal{F}^\phi} \inf_{f' \in \mathcal{F}^\phi} \|f' - \mathcal{T}^{\pi_e} f\|_\infty = 0$.*

Proof. We first define π_e-bisimulation Castro (2019). Note that Castro (2019) considered only state abstractions, while we consider state-action abstractions. ϕ is considered a π_e-bisimulation abstraction if it induces a mapping between $\mathcal{X}$ and $\mathcal{X}^\phi$ such that for any $x, y \in \mathcal{X}$ such that $x, y \in \phi(x)$, we have:

1. $r(x) = r(y)$

2. $\forall x^\phi \in \mathcal{X}^\phi, \sum_{x' \in x^\phi} P^{\pi_e}(x'|x) = \sum_{x' \in x^\phi} P^{\pi_e}(x'|y)$

Given our π_e-bisimulation abstraction function ϕ, we can group state-actions actions according to its definition above. Once we have this grouping, according to Definition 9, ϕ induces a piece-wise constant (PWC) function class $\mathcal{F}^\phi$. Note that by definition of ϕ we have:

$$\epsilon_r := \max_{x_1, x_2 : \phi(x_1) = \phi(x_2)} |r(x_1) - r(x_2)| = 0$$

$$\epsilon_p := \max_{x_1, x_2 : \phi(x_1) = \phi(x_2)} \left| \sum_{x' \in x^\phi} P^{\pi_e}(x'|x_1) - \sum_{x' \in x^\phi} P^{\pi_e}(x'|x_2) \right| = 0, \forall x^\phi \in \mathcal{X}^\phi.$$

Once we have ϕ, we consider the following to show Bellman completeness. Our proof closely follows the proof of Proposition 20 from Chen & Jiang (2019). First recall the definition of Bellman completeness from Definition 8: $\forall f \in \mathcal{F}, \forall \mathcal{T}^{\pi_e} f \in \mathcal{G}$, $\sup_{f \in \mathcal{F}} \inf_{g \in \mathcal{G}} \|g - \mathcal{T}^{\pi_e} f\|_\infty = 0$. Given that the smallest value $\forall f \in \mathcal{F}, \forall \mathcal{T}^{\pi_e} f \in \mathcal{G}$, $\sup_{f \in \mathcal{F}} \inf_{g \in \mathcal{G}} \|g - \mathcal{T}^{\pi_e} f\|_\infty$ can take on is zero, we will prove our claim by showing that $\forall f \in \mathcal{F}, \forall \mathcal{T}^{\pi_e} f \in \mathcal{G}$, $\sup_{f \in \mathcal{F}} \inf_{g \in \mathcal{G}} \|g - \mathcal{T}^{\pi_e} f\|_\infty$ is upper-bounded by zero when ϕ is a π_e-bisimulation.

We will prove the upper bound by showing that there exists a function $f' \in \mathcal{F}^\phi$ such that $\|f' - \mathcal{T}^{\pi_e} f\|_\infty \leq 0$, which implies that $\inf_{f' \in \mathcal{F}^\phi} \|f' - \mathcal{T}^{\pi_e} f\|_\infty \leq 0$.

We now construct such a $f' \in \mathcal{F}^\phi$. We first define the following terms for a given abstract state-action $x^\phi \in \mathcal{X}^\phi$: $x_+ := \arg\max_{x \in \phi^{-1}(x^\phi)} (\mathcal{T}^{\pi_e} f)(x)$ and $x_- := \arg\min_{x \in \phi^{-1}(x^\phi)} (\mathcal{T}^{\pi_e} f)(x)$. We can then define f' as follows:

$$f'(x) := \frac{1}{2}((\mathcal{T}^{\pi_e} f)(x_+) + (\mathcal{T}^{\pi_e} f)(x_-)), \forall x \in x^\phi.$$

And since this holds true for $\forall x \in x^\phi$, f_1' is piece-wise constant function. We can then upper bound $\|f' - \mathcal{T}^{\pi_e} f\|_\infty$ as follows:

$$
\begin{aligned}
& f_1'(x) - (\mathcal{T}^{\pi_e} f)(x) \\
&\leq \frac{1}{2}((\mathcal{T}^{\pi_e} f)(x_+) + (\mathcal{T}^{\pi_e} f)(x_-)) - (\mathcal{T}^{\pi_e} f)(x_-) \\
&= \frac{1}{2}((\mathcal{T}^{\pi_e} f)(x_+) - (\mathcal{T}^{\pi_e} f)(x_-)) \\
&= \frac{1}{2}(r(x_+) + \gamma \mathbb{E}_{x'_+ \sim P^{\pi_e}(x_+)}[f^{\pi_e}(x'_+)] - r(x_-) - \gamma \mathbb{E}_{x'_- \sim P^{\pi_e}(x_-)}[f^{\pi_e}(x'_-)]) \\
&\leq \frac{\gamma}{2} \left| \mathbb{E}_{x'_+ \sim P^{\pi_e}(x_+)}[f^{\pi_e}(x'_+)] - \mathbb{E}_{x'_- \sim P^{\pi_e}(x_-)}[f^{\pi_e}(x'_-)] \right| && (1) \\
&= \frac{\gamma}{2} \left| \sum_{x' \in \mathcal{X}} [f^{\pi_e}(x')(P^{\pi_e}(x'|x_+) - P^{\pi_e}(x'|x_-))] \right| \\
&= \frac{\gamma}{2} \left| \sum_{x^\phi \in \mathcal{X}^\phi} \left(\sum_{x' \in x^\phi} f^{\pi_e}(x') P^{\pi_e}(x'|x_+) - \sum_{x' \in x^\phi} f^{\pi_e}(x') P^{\pi_e}(x'|x_-) \right) \right| \\
&= \frac{\gamma}{2} \left| \sum_{x^\phi \in \mathcal{X}^\phi} f^{\pi_e}(x^\phi) \left(\sum_{x' \in x^\phi} P^{\pi_e}(x'|x_+) - \sum_{x' \in x^\phi} P^{\pi_e}(x'|x_-) \right) \right| && (2) \\
&= \frac{\gamma}{2} \left| \sum_{x^\phi \in \mathcal{X}^\phi} f^{\pi_e}(x^\phi) \left(\Pr(x^\phi|x_+) - \Pr(x^\phi|x_-) \right) \right| && (3) \\
&\leq \frac{\gamma}{2} \left\| \Pr(x^\phi|x_+) - \Pr(x^\phi|x_-) \right\|_1 \cdot \|f^{\pi_e}(x^\phi)\|_\infty && (4) \\
&\leq 0 && \epsilon_p = 0
\end{aligned}
$$

where Pr denotes probability, (1) is due to $\max_{x_1,x_2:\phi(x_1)=\phi(x_2)} |r(x_1) - r(x_2)| = 0$, (2) is due to $f^{\pi_e}(x^\phi) = f^{\pi_e}(x), \forall x \in x^\phi$ since PWC, (3) is due to $\Pr(x^\phi|x) = \sum_{x' \in x^\phi} P^{\pi_e}(x'|x)$, and (4) is due to Hölder's, $\|f(g)g(x)\|_1 \leq \|f(x)\|_1 \|g(x)\|_\infty$.

Similarly, we can show the other way around: $(\mathcal{T}^{\pi_e} f)(x) - f_1'(x) \leq 0$ by giving the symmetric argument starting with $(\mathcal{T}^{\pi_e} f)(x) - f_1'(x) \leq (\mathcal{T}^{\pi_e} f)(x_+) - \frac{1}{2}((\mathcal{T}^{\pi_e} f)(x_+) + (\mathcal{T}^{\pi_e} f)(x_-))$. Therefore, when ϕ is a π_e-bisimulation, we have $\sup_{f \in \mathcal{F}^\phi} \inf_{f' \mathcal{F}^\phi} \|f' - \mathcal{T}^{\pi_e} f\|_\infty = 0$. □

Lemma 4. *Define the matrix $K_1 \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{X}|}$ such that each entry is the short-term similarity, k_1, between every pair of state-actions, i.e., $K_1(s_1, a_1; s_2, a_2) := 1 - \frac{1}{|r_{max} - r_{min}|} |r(s_1, a_1) - r(s_2, a_2)|$. Then K_1 is a positive semidefinite matrix.*

Proof. Proposition 2.21 from Paulsen & Raghupathi (2016) states that any kernel k is positive semidefinite if it takes the form: $k(a, b) = \min\{a, b\}$ where $a, b \in [0, \infty)$.

First, recall that $r(s, a) \in [-1, 1]$, we then have each entry in the K_1 matrix of the following kernel form $K_1(x, y) = 1 - \frac{1}{2}|x - y|$. We can then re-write k_1 as follows:

$$\begin{aligned} k_1(x, y) &= 1 - \frac{1}{2}|x - y| \\ &= 1 + \frac{1}{2}\min\{-x, -y\} + \frac{1}{2}\min\{x, y\} \\ &= \underbrace{\frac{1}{2}\min\{1 - x, 1 - y\}}_{k_a} + \underbrace{\frac{1}{2}\min\{1 + x, 1 + y\}}_{k_b}. \end{aligned}$$

That is,

$$k_1(x, y) = k_a(x, y) + k_b(x, y).$$

Since $x \in [-1, 1]$, each term in the min function is non-negative. Thus, k_a and k_b are both positive semidefinite kernels, which means k_1 is also a positive semidefinite kernel. We then have that K_1 is a positive semidefinite matrix. □

Lemma 5. *Given a finite set $\mathcal{X}$ and a kernel k defined on $\mathcal{X}$, let $K = (k(x, y))_{x, y \in \mathcal{X}} \in \mathbb{R}^{|\mathcal{X}| \times |\mathcal{X}|}$ be the corresponding kernel matrix. If K is full-rank and MMD$(k)(p, q) = 0$ for two probability distributions p and q on $\mathcal{X}$, then $p = q$.*

Proof. From Gretton et al. (2012), we have the definition of MMD between two probability distributions p, q given kernel k:

$$\text{MMD}(k)(p, q) := \|\mathbb{E}_{x \sim p}[k(x, \cdot)] - \mathbb{E}_{x \sim q}[k(x, \cdot)]\|_{\mathcal{H}_k}.$$

Now when MMD$(k)(p, q) = 0$, we have:

$$0 = \|\mathbb{E}_{x \sim p}[k(x, \cdot)] - \mathbb{E}_{x \sim q}[k(x, \cdot)]\|_{\mathcal{H}_k},$$

which implies

$$0 = \|\mathbb{E}_{x \sim p}[k(x, \cdot)] - \mathbb{E}_{x \sim q}[k(x, \cdot)]\|_2$$

since all norms are equivalent in a finite-dimensional Hilbert space. With p and q viewed as vectors in $\mathbb{R}^{|\mathcal{X}|}$, the above equality means

$$0 = \|Kp - Kq\|_2.$$

Hence, $K(p - q) = 0$. Since K is full rank by assumption, we conclude that $p = q$. □

Lemma 6. *Suppose we have a reproducing kernel k defined on the finite space $\mathcal{X}$, which produces a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_k$, with the induced distance function d such that $d(x, y) = k(x, x) + k(y, y) - 2k(x, y), \forall x, y \in \mathcal{X}$. When $d(x, y) = 0$, then $k(x, \cdot) = k(y, \cdot)$.*

Proof. When $d(x, y) = 0$, we have $2k(x, y) = k(x, x) + k(y, y)$. Therefore, we the following equalities:

$$
\begin{aligned}
k(x,x) + k(y,y) &= 2k(x,y) \\
k(x,x) - k(x,y) &= k(x,y) - k(y,y) \\
\langle k(x,\cdot), k(x,\cdot)\rangle_{\mathcal{H}_k} - \langle k(x,\cdot), k(y,\cdot)\rangle_{\mathcal{H}_k} &= \langle k(x,\cdot), k(y,\cdot)\rangle_{\mathcal{H}_k} - \langle k(y,\cdot), k(y,\cdot)\rangle_{\mathcal{H}_k} \quad (1) \\
\langle k(x,\cdot), k(x,\cdot) - k(y,\cdot)\rangle_{\mathcal{H}_k} &= \langle k(x,\cdot) - k(y,\cdot), k(y,\cdot)\rangle_{\mathcal{H}_k} \quad (2) \\
\langle k(x,\cdot), k(x,\cdot) - k(y,\cdot)\rangle_{\mathcal{H}_k} &= \langle k(y,\cdot), k(x,\cdot) - k(y,\cdot)\rangle_{\mathcal{H}_k} \quad (3) \\
\implies k(x,\cdot) &= k(y,\cdot)
\end{aligned}
$$

where (1), (2), and (3) are is due to RKHS definition, linearity of inner product, and symmetry of inner product respectively. □

Proposition 5. *Let $x_1, \dots, x_n \in (0, \infty)$ be n distinct and strictly positive numbers. Let $K \in \mathbb{R}^{n\times n}$ be the matrix with entries $K_{ij} = \min\{x_i, x_j\}$. Then K is a positive definite matrix.*

Proof. By Proposition 2.21 in Paulsen & Raghupathi (2016), the matrix K is positive semidefinite, so we only need to show that K is full rank. WLOG assume that $0 < x_1 < x_2 < \cdots < x_n$. We prove by induction on n. The base case with $n = 1$ clearly holds. Suppose the claim holds for $n - 1$ numbers. Now consider n numbers. Let $\alpha = (\alpha_1, \dots, \alpha_n)^\top \in \mathbb{R}^n$. It suffices to show that $K\alpha = 0$ implies $\alpha = 0$. We write K in block matrix form as

$$
K = x_1 J_n + \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & x_2 - x_1 & \cdots & x_2 - x_1 \\ \vdots & \cdots & \ddots & \cdots \\ 0 & x_2 - x_1 & \cdots & x_n - x_1 \end{bmatrix} = x_1 \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \mathbf{1} & \mathbf{1} & \cdots & \mathbf{1} \end{bmatrix} + \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \mathbf{0} & \mathbf{u}_2 & \cdots & \mathbf{u}_n \end{bmatrix},
$$

where J_n is the n-by-n all one matrix, $\mathbf{1} \in \mathbb{R}^{n-1}$ the all one vector, $\mathbf{0} \in \mathbb{R}^{n-1}$ the all zero vector, and $\mathbf{u}_i \in \mathbb{R}^{n-1}, i = 2, \dots, n$. It follow that

$$
0 = K\alpha = \begin{bmatrix} x_1 \sum_{i=1}^n \alpha_i \\ \left(x_1 \sum_{i=1}^n \alpha_i\right) \mathbf{1} + \sum_{i=2}^n \alpha_i \mathbf{u_i} \end{bmatrix},
$$

that is,

$$
x_1 \sum_{i=1}^n \alpha_i = 0, \tag{7}
$$

$$
\left(x_1 \sum_{i=1}^n \alpha_i\right) \mathbf{1} + \sum_{i=2}^n \alpha_i \mathbf{u_i} = \mathbf{0}. \tag{8}
$$

Plugging equation (7) into equation (8), we get $\sum_{i=2}^n \alpha_i \mathbf{u_i} = \mathbf{0}$. By the induction hypothesis, the $(n - 1)$-by-$(n - 1)$ matrix

$$
[\mathbf{u}_2 \quad \cdots \quad \mathbf{u}_n] = \left[\, \min\{x_i - x_1, x_j - x_1\} \,\right]_{i,j=2,\dots,n}
$$

has full rank since the $(n - 1)$ numbers $x_2 - x_1, \dots, x_n - x_1$ are distinct and strictly positive. Therefore, we must have $\alpha_2 = \cdots = \alpha_n = 0$. Plugging back into equation (7) and using $x_1 > 0$, we obtain $\alpha_1 = 0$. □

B.3 Main krope Theoretical Results

We now present the main theoretical contributions of our work.

Theorem 1. *If Φ is a KROPE representation as defined in Definition 3, then the spectral radius of $(\mathbb{E}[\Phi^\top \Phi]))^{-1}\mathbb{E}[\gamma \Phi^\top P^{\pi_e} \Phi]$ is less than 1. That is, Φ stabilizes LSPE.*

Proof. Recall from Definition 3, we have:

$$
\mathbb{E}[\Phi\Phi^\top] = K_1 + \gamma \mathbb{E}[P^{\pi_e}\Phi(P^{\pi_e}\Phi)^\top],
$$

where $K_1 \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$ such that each entry is the short-term similarity, k_1, between every pair of state-actions i.e. $K_1(s_1, a_1; s_2, a_2) := 1 - \frac{|r(s_1,a_1)-r(s_2,a_2)|}{|r_{\max}-r_{\min}|}$.

From this definition, we can proceed by left and right multiplying $\Phi^\top$ and Φ respectively to get:

$$\mathbb{E}[\Phi^\top\Phi\Phi^\top\Phi] = \mathbb{E}[\Phi^\top K_1\Phi] + \gamma\mathbb{E}[\Phi^\top P^{\pi_e}\Phi(P^{\pi_e}\Phi)^\top\Phi].$$

Notice that $B := \mathbb{E}[\Phi^\top\Phi]$ is the feature covariance matrix and $C := \mathbb{E}[\Phi^\top P^{\pi_e}\Phi]$ is the cross-covariance matrix. By making the appropriate substitutions, we get:

$$BB^\top = \mathbb{E}[\Phi^\top K_1\Phi] + \gamma CC^\top.$$

We can then left and right multiply by B^{-1} and $B^{-\top}$ to get the following where $L := \gamma B^{-1}C$:

$$I = B^{-1}\mathbb{E}[\Phi^\top K_1\Phi]B^{-\top} + \frac{1}{\gamma}LL^\top.$$

Rearranging terms gives

$$I - \frac{1}{\gamma}LL^\top = B^{-1}\mathbb{E}[\Phi^\top K_1\Phi]B^{-\top}.$$

From Lemma 4, we know that K_1 is positive semidefinite, which means that $B^{-1}\mathbb{E}[\Phi^\top K_1\Phi]B^{-\top}$ is also positive semidefinite. Therefore, the eigenvalues of LHS above must also be greater than or equal to zero. Letting λ be the eigenvalue of L, we know that that the following must hold:

$$1 - \frac{\lambda^2}{\gamma} \geq 0 \implies |\lambda| \leq \sqrt{\gamma}.$$

Since $\gamma < 1$, the spectral radius of $L = (\mathbb{E}[\Phi^\top\Phi])^{-1}(\gamma\mathbb{E}[\Phi^\top P^{\pi_e}\Phi])$ is always less than 1. Thus, KROPE representations are stable. Finally, since KROPE representations are stable and due to Proposition 1, KROPE representations stabilize LSPE. □

Theorem 2. *Let $\phi : \mathcal{X} \to \mathcal{X}^\phi$ be the state-action abstraction induced by grouping state-actions $x, y \in \mathcal{X}$ such that if $d_{\text{KROPE}}(x, y) = 0$, then $\phi(x) = \phi(y), \forall x, y \in \mathcal{X}$. Then ϕ is Bellman complete if the abstract reward function $r^\phi : \mathcal{X}^\phi \rightarrowtail (-1, 1)$ is injective (i.e., distinct abstract rewards).*

Proof. Our proof strategy is to show that the abstraction function ϕ due to KROPE is a π_e-bisimulation, which implies it is Bellman complete due to Proposition 4.

According to Propsition 3, $d_{\text{KROPE}}(x, y) = |r(x) - r(y)| + \gamma\text{MMD}(k^{\pi_e})(P^{\pi_e}(\cdot|x), P^{\pi_e}(\cdot|y))$. When $d_{\text{KROPE}}(x, y) = 0$ for any two state-actions, it implies that $r(x) = r(y)$ and $\text{MMD}(k^{\pi_e})(P^{\pi_e}(\cdot|x), P^{\pi_e}(\cdot|y)) = 0$.

For ϕ to be a π_e-bisimulation, we need $\forall x^\phi \in \mathcal{X}^\phi, \sum_{x'\in x^\phi} P^{\pi_e}(x'|x) = \sum_{x'\in x^\phi} P^{\pi_e}(x'|y)$ to be true for any $x, y \in \mathcal{X}$ such that $\phi(x) = \phi(y)$. While $\text{MMD}(k^{\pi_e})(P^{\pi_e}(\cdot|x), P^{\pi_e}(\cdot|y)) = 0$, it is possible that $P^{\pi_e}(\cdot|x) \neq P^{\pi_e}(\cdot|y)$. However, as we will show, under the assumption that the abstract rewards r^ϕ are distinct $\forall x^\phi \in \mathcal{X}^\phi$, we do have $\forall x^\phi \in \mathcal{X}^\phi, \sum_{x'\in x^\phi} P^{\pi_e}(x'|x) = \sum_{x'\in x^\phi} P^{\pi_e}(x'|y)$. Before we proceed, we make the following technical assumption on the reward function: $r(x) \in (-1, 1), \forall x \in \mathcal{X}$. The exclusion of the rewards -1 and 1 allows us to use Proposition 5 to show that the KROPE kernel is positive definite instead of positive semi-definite.

Once we group state-actions $x, y \in \mathcal{X}$ together such that $d_{\text{KROPE}}(x, y) = 0$, we have the corresponding abstraction function $\phi : \mathcal{X} \to \mathcal{X}^\phi$. Accordingly, ϕ induces a Markov reward process, $\mathcal{M}^\phi := \langle\mathcal{X}^\phi, r^\phi, P^\phi, \gamma\rangle$ where r^ϕ is the abstract reward function $r^\phi : \mathcal{X}^\phi \to (-1, 1)$ and P^ϕ is the transition dynamics on the abstract MRP i.e. $P^\phi(\cdot|x^\phi)$. We can also consider the abstract KROPE kernel, $k^\phi(x^\phi, y^\phi)$, which measures the KROPE relation on $\mathcal{X}^\phi$. Note that all these quantities are a function of π_e. We drop the notation for clarity. By this construction, we have:

$$\begin{aligned} r^\phi(x^\phi) &= r(x), \forall x \in x^\phi && \text{Since all rewards are equal within } x^\phi \\ k^\phi(x^\phi, \cdot) &= k(x, \cdot), \forall x \in x^\phi && \text{Lemma 6} \end{aligned}$$

Now, under the assumption that all abstract rewards $r^\phi(x^\phi)$ are distinct $\forall x^\phi \in \mathcal{X}^\phi$, we have that the kernel matrix $K^\phi \in \mathbb{R}^{\mathcal{X}^\phi\times\mathcal{X}^\phi}$ where each entry $k^\phi(x^\phi, y^\phi)$ is positive definite. To see this fact, consider that:

$$k^\phi(x^\phi, y^\phi) = k_1^\phi(x^\phi, y^\phi) + \gamma\mathbb{E}_{X^\phi\sim P^\phi(\cdot|x^\phi), Y^\phi\sim P^\phi(\cdot|y^\phi)}[k^\phi(X^\phi, Y^\phi)], \tag{9}$$

where $k_1^\phi(x^\phi, y^\phi) := 1 - \frac{1}{r^\phi_{\text{max}} - r^\phi_{\text{min}}}|r^\phi(x^\phi) - r^\phi(y^\phi)|$. From Lemma 4, we know that k_1^ϕ is positive semidefinite. However, under the assumption that all abstract rewards r^ϕ are distinct, Proposition 5 tells us that k_1^ϕ is positive definite. Given that $k^\phi(x^\phi, y^\phi)$ (Equation (9)) is just a summation of positive definite kernels, k^ϕ is positive definite, which means K^ϕ is positive definite.

We now consider when the MMD is zero. Again, by construction, we have the following when $\text{MMD}(k^{\pi_e})(P^{\pi_e}(\cdot|x), P^{\pi_e}(\cdot|y)) = 0$. For clarity, we use k instead of k^{π_e}.

$$
\begin{aligned}
0 &= \|\mathbb{E}_{X'\sim P^{\pi_e}(\cdot|x)}[k(X',\cdot)] - \mathbb{E}_{X'\sim P^{\pi_e}(\cdot|y)}[k(X',\cdot)]\|_{\mathcal{H}_k} \\
&= \|\mathbb{E}_{X'\sim P^{\pi_e}(\cdot|x)}[k(X',\cdot)] - \mathbb{E}_{X'\sim P^{\pi_e}(\cdot|y)}[k(X',\cdot)]\|_2 \\
&= \|\sum_{x'\in\mathcal{X}} P^{\pi_e}(x'|x)k(x',\cdot) - \sum_{x'\in\mathcal{X}} P^{\pi_e}(x'|y)k(x',\cdot)\|_2 \\
&= \left\| \sum_{x^\phi\in\mathcal{X}^\phi}\sum_{x'\in x^\phi} P^{\pi_e}(x'|x)k(x',\cdot) - \sum_{x^\phi\in\mathcal{X}^\phi}\sum_{x'\in x^\phi} P^{\pi_e}(x'|y)k(x',\cdot) \right\|_2 \\
&= \left\| \sum_{x^\phi\in\mathcal{X}^\phi} k^\phi(x^\phi,\cdot) \sum_{x'\in x^\phi} P^{\pi_e}(x'|x) - \sum_{x^\phi\in\mathcal{X}^\phi} k^\phi(x^\phi,\cdot) \sum_{x'\in x^\phi} P^{\pi_e}(x'|y) \right\|_2 && (1) \\
&= \left\| \sum_{x^\phi\in\mathcal{X}^\phi} k^\phi(x^\phi,\cdot)\Pr(x^\phi|x) - \sum_{x^\phi\in\mathcal{X}^\phi} k^\phi(x^\phi,\cdot)\Pr(x^\phi|y) \right\|_2 && \Pr \text{ denotes probability}
\end{aligned}
$$

where (1) is due to $k^\phi(x^\phi,\cdot) = k(x,\cdot), \forall x \in x^\phi$ From above, we can see that the kernel and probability distributions are over $\mathcal{X}^\phi$. In matrix notation, we can write the above as follows where $p^\phi := \Pr(\cdot|x)$ and $q^\phi := \Pr(\cdot|y)$ are viewed as probability distribution vectors in $\mathbb{R}^{|\mathcal{X}^\phi|}$.

$$
\begin{aligned}
0 &= \left\|K^\phi p^\phi - K^\phi q^\phi\right\|_2 \\
&\implies p^\phi = q^\phi && \text{since } K^\phi \text{ is positive definite, from Lemma 5.}
\end{aligned}
$$

We thus have $\forall x^\phi \in \mathcal{X}^\phi, \sum_{x'\in x^\phi} P^{\pi_e}(x'|x) = \sum_{x'\in x^\phi} P^{\pi_e}(x'|y)$ to be true for any $x, y \in \mathcal{X}$ such that $\phi(x) = \phi(y)$. Given this condition holds true and $r(x) = r(y), \forall x, y \in \mathcal{X}$ such that $\phi(x) = \phi(y)$, ϕ is a π_e-bisimulation. From Proposition 4 we then have that ϕ is Bellman complete.

□

C Empirical Details

In this section, we provide specific details on the empirical setup and additional results.

C.1 Empirical Setup

General Training Details. In all the continuous state-action experiments, we use a neural network with 1 layer and 1024 neurons using RELU activation function and layernorm to represent the encoder $\phi : \mathcal{X} \to \mathbb{R}^d$ (Gallici et al., 2024). We use mini-batch gradient descent to train the network with mini-batch sizes of 2048 and for 500 epochs, where a single epoch is a pass over the full dataset. We use the Adam optimizer with learning rate $\{1e^{-5}, 2e^{-5}, 5e^{-5}\}$ and weight decay $1e^{-2}$. The target network is updated with a hard update after every epoch. The output dimension d is $\{|\mathcal{X}|/4, |\mathcal{X}|/2, 3|\mathcal{X}|/4\}$, where $|\mathcal{X}|$ is the dimension of the original state-action space of the environment. All our results involve analyzing this learned ϕ. Since FQE outputs a scalar, we add a linear layer on top of the d-dimensional vector to output a scalar. The entire network is then trained end-to-end. The discount factor is $\gamma = 0.99$. The auxiliary task weight with FQE for all representation learning algorithms is $\alpha = 0.1$. When using LSPE for OPE, we invert the covariance matrix by computing the pseudoinverse.

In the tabular environments, we use a similar setup as above. The only changes are that we use a linear network with a bias component but no activation function and fix the learning rate to be $1e^{-3}$. For experiment in Appendix C.3.1, $\alpha = 0.8$. We refer the reader pseudo-code in Appendix A.

Evaluation Protocol: OPE Error . As noted earlier, we measure OPE error by measuring MSVE. To ensure comparable and interpretable values, we normalize the MSVE by dividing with $\text{MSVE}[q^{\text{RAND}}] := \mathbb{E}_{(S,A)\sim\mathcal{D}}[(q^{\text{RAND}}(S,A) - q^{\pi_e}(S,A))^2]$, where q^{RAND} is the action-value function of a random-policy. Similarly, in the continuous state-action environments, we normalize by $\text{MSVE}[q^{\text{RAND}}] := \mathbb{E}_{S_0\sim d_0, A_0\sim\pi_e}[(q^{\text{RAND}}(S_0,A_0) - q^{\pi_e}(S_0,A_0))^2]$. Values less than one mean that the algorithm estimates the true performance of π_e better than a random policy.

Evaluation Protocol: Realizability Error. In tabular experiments, we normalize the realizability error. After solving the least-squares problem $\epsilon := \|\Phi\hat{w} - q^{\pi_e}\|_2^2$, where $\hat{w} := \arg\min_w \|\Phi w - q^{\pi_e}\|_2^2$. We divide ϵ by $\frac{1}{|\mathcal{X}|}\sum_i |q^{\pi_e}(x_i)|$ and plot this value.

Pearson Correlation. The formula for the Pearson correlation used in the main experiments is:

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2}\sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}}$$

where $\bar{x}$ and $\bar{y}$ are the means of all the x_i's and y_i's respectively.

Custom Datasets. We generated the datasets by first training policies in the environment using SAC (Haarnoja et al., 2018) and recording the trained policies during the course of training. For each environment, we select 3 policies, where each contributes equally to generate a given dataset. We set π_e to be one of these policies. The expected discounted return of the policies and datasets for each domain is given in Table 1 ($\gamma = 0.99$). In all environments, $\pi_e = \pi_b^1$ (see Table 1). The values for the evaluation and behavior policies were computed by running each for 300 unbiased Monte Carlo rollouts, which was more than a sufficient amount for the estimate to converge. This process results in total of 4 datasets, each of which consisted of 100K transitions.

Environments	π_e	π_b^1	π_b^2
CartPoleSwingUp	50	20	5
FingerEasy	100	71	32
HalfCheetah	51	27	2
WalkerStand	90	55	40

Table 1: Policy values of the target policy and behavior policy on DM-control (Tassa et al., 2018).

D4RL Datasets. Due to known discrepancy issues between newer environments of gym[1], we generat our datasets instead of using the publicly available ones. To generate the datasets, we use the publicly available policies [2]. For each domain, the expert (and target policy) was the 10th (last policy) from training. The medium (and behavior policy) was the 5th policy. We added a noise of 0.1 to the policies. The values for the evaluation and behavior policies were computed by running each for 300 unbiased Monte Carlo rollouts , which was more than a sufficient amount for the estimate to converge. We set $\gamma = 0.99$. We evaluate on the Cheetah, Walker, and Hopper domains. This generation process for three environments, led to 9 datasets, each of which consisted of 100K transitions.

C.2 Baselines

We provide details of the baselines in this section.

BCRL. Unlike the other algorithms, BCRL is not used as an auxiliary loss with FQE (Chang et al., 2022). We use the same learning rates as mentioned above for $\{1e^{-5}, 2e^{-5}, 5e^{-5}\}$ when training

[1] https://github.com/Farama-Foundation/D4RL/tree/master
[2] https://github.com/google-research/deep_ope

ϕ. As suggested by prior work, self-predictive algorithms such BCRL work well when network that outputs the predicted next state-action is trained at a faster rate (Tang et al., 2023; Grill et al., 2020). Accordingly, we set its learning rate to be $1e^{-4}$. For BCRL-EXP, which involves the log determinant regularizer, we set this coefficient to $1e^{-2}$. BCRL's hyperparameters are: the learning rates for ϕ, M, and ρ; the output dimension of ϕ; and the log determinant coefficient (see Equation (10)).

DR3. The DR3 regularizer minimizes the total feature co-adaptation by adding the term $\sum_{(s,a,s')\in\mathcal{D},a'\sim\pi_e}\phi(s,a)^\top\phi(s',a')$ as an auxiliary task to the main FQE loss (Kumar et al., 2021). Ma et al. (2024) introduced an improvement to this auxiliary loss by suggesting that the absolute value of the feature co-adaptation be minimized, i.e., $\sum_{(s,a,s')\in\mathcal{D},a'\sim\pi_e}|\phi(s,a)^\top\phi(s',a')|$. e use $\alpha = 0.1$ as its auxiliary task weight. Absolute DR3's only hyperparameters are the auxiliary task weight α and the ϕ output dimension.

BEER. He et al. (2024) introduced an alternative regularizer to DR3 rank regularizer since they suggested that the minimization of the unbounded feature co-adaptation can undermine performance. They introduced their bounded rank regularizer BEER (see Equation (12) in He et al. (2024)). BEER introduces only the auxiliary task weight α as the additional hyperparameter.

KROPE. KROPE's only hyperparameters are the output dimension of ϕ and the learning rate of the KROPE learning algorithm.

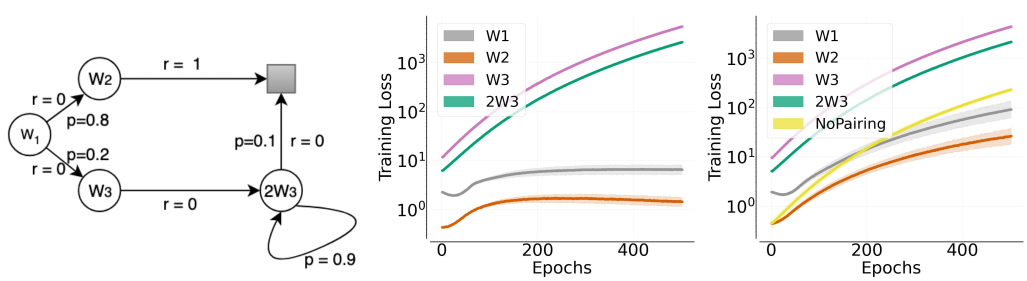
C.3 Additional Results

In this section, we include additional empirical results.

C.3.1 Understanding the Stability of the KROPE Learning Procedure

In this section analyze how the deadly triad affects KROPE learning procedure. Since KROPE's learning algorithm is a semi-gradient method, it is susceptible to instability (Tsitsiklis & Van Roy, 1997). As such, these experiments aim to understand when we might expect KROPE to diverge and when it might mitigate the divergence of FQE.

We conduct our experiments on the Markov reward process in Figure 5(a), which was introduced by Feng et al. (2019). The MRP consists of 4 non-terminal states, 1 terminal state (the box), and only 1 action. The value function estimate is linear in the weight $w = [w_1, w_2, w_3]$, so the native features of the states are $[1, 0, 0]$, $[0, 1, 0]$, $[0, 0, 2]$, and $[0, 0, 1]$ (clockwise starting from left-most). We set $\gamma = 1$, so the optimal weights are $w^* = [0.8, 1, 0]$. In this setup, we say a transition is a *bad transition* if the bootstrapping target is a moving target for the current state (Asadi et al., 2024). For example, the transition from w_3 to $2w_3$ is a bad transition since updates made to w_3 may move $2w_3$ further away. When this transition is sampled at a frequency that is different from the on-policy distribution, algorithms such as TD, LSPE, and FQE tend to diverge (Asadi et al., 2024).



(a) Divergence Counterexample (b) KROPE Training loss (c) FQE+KROPE Training loss

Figure 5: Figure 5(a): Markov reward process counterexample designed to illustrate divergence; r denotes the rewards and p denotes the probability of transition (Feng et al., 2019). Figures 5(b) and 5(c): KROPE training loss and FQE+KROPE training loss vs. epochs respectively when different datasets are paired with $\mathcal{D}_1$; results are averaged over 20 trials, shaded region is the 95% confidence interval, and lower is better.

To better understand the stability of the learning procedures, we design the following experiment. We start with two datasets $\mathcal{D}_1$ and $\mathcal{D}_2$. $\mathcal{D}_1$ consists of 2000 on-policy transitions and 5000 off-policy bad transitions from w_3 to $2w_3$, while $\mathcal{D}_2$ consists of only 2000 on-policy transitions. We then consider four different variations of $\mathcal{D}_2$: $\mathcal{D}_2^{w_1}$, $\mathcal{D}_2^{w_2}$, $\mathcal{D}_2^{w_3}$, and $\mathcal{D}_2^{2w_3}$, where each variation denotes a dataset where we add 5000 off-policy transitions from the specified state to $\mathcal{D}_2$. With this setup, our goal is understand which dataset pairing with $\mathcal{D}_1$: $(\mathcal{D}_1, \mathcal{D}_2^{w_1})$, $(\mathcal{D}_1, \mathcal{D}_2^{w_2})$, $(\mathcal{D}_1, \mathcal{D}_2^{w_3})$, and $(\mathcal{D}_1, \mathcal{D}_2^{2w_3})$ increases the susceptibility of KROPE to divergence. Recall that at each training step, KROPE samples *pairs* of transitions. Thus, in this setup, KROPE will sample one transition from $\mathcal{D}_1$ and the other transition from the other dataset.

In Figures 5(b) and 5(c), we show the training loss of KROPE only and FQE+KROPE with KROPE as an auxiliary task. In both cases, the sampled state-actions are fed into a linear network encoder with a bias component and no activation function which outputs a $d = 3$ representation. In case of FQE, this representation is then fed into a linear function to output the scalar value. From Figure 5(b), we find that even though $\mathcal{D}_1$ consists of mostly bad transitions, if KROPE also samples from $\mathcal{D}_2^{w_1}$ or $\mathcal{D}_2^{w_2}$, its divergence is mitigated. However, as expected, if KROPE uses $\mathcal{D}_2^{w_3}$ or $\mathcal{D}_2^{2w_3}$, KROPE diverges since the pairing of samples from $(\mathcal{D}_1, \mathcal{D}_2^{w_3})$, and $(\mathcal{D}_1, \mathcal{D}_2^{2w_3})$ leads to KROPE chasing a moving bootstrapped target, resulting in divergence. Building upon this insight, we can then better understand when KROPE can mitigate FQE's divergence. From Figure 5(c), we see the expected result that when FQE uses $\mathcal{D}_1$ only, it diverges since $\mathcal{D}_1$ consists of many bad transitions. However, if FQE uses KROPE as an auxiliary loss, KROPE may either reduce the divergence ($\mathcal{D}_2^{w_1}$ and $\mathcal{D}_2^{w_2}$) or worsen it ($\mathcal{D}_2^{w_3}$ and $\mathcal{D}_2^{2w_3}$) depending on which dataset is paired with $\mathcal{D}_1$. While in practice, it is unclear to determine which situation is more likely, this result sheds light on the fact that the KROPE learning procedure can potentially mitigate the divergence of FQE's learning procedure depending on which *pairs* of transitions are sampled.

Takeaway #3: Understanding KROPE Divergence

Depending on which *pairs* of transitions are sampled, KROPE can potentially mitigate or worsen the divergence of FQE.

C.3.2 Offline Policy Evaluation on D4RL Datasets

In this section, we present the offline policy evaluation results on the D4RL datasets. The setup is the same as the one used in Section 4. We present the results in Figure 6.

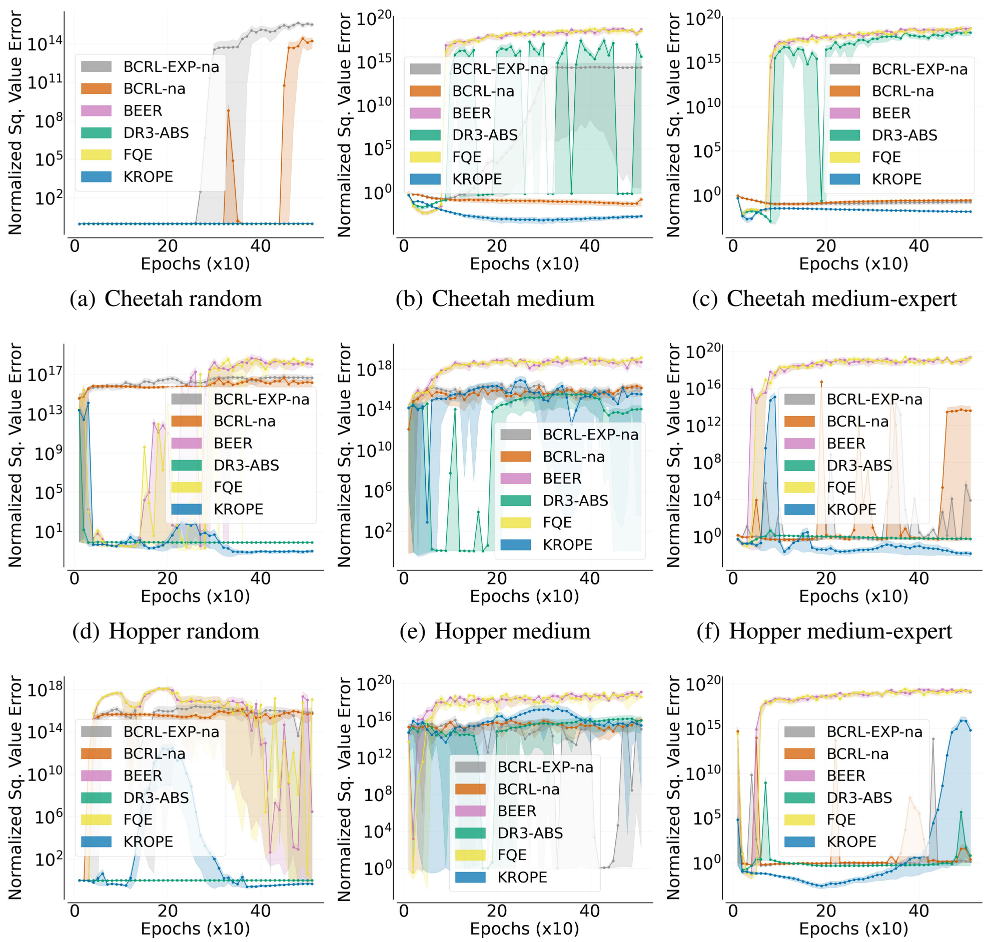
Qualitatively, we reach the similar conclusions: KROPE is effective in producing stable and accurate OPE estimates. However, in $3/9$ instances, KROPE does diverge. This divergence is likely related to the discussion in Section 5 and Appendix C.3.1. Recall that KROPE is a semi-gradient method, which does not optimize any objective function and is susceptible to divergence (Feng et al., 2019; Sutton & Barto, 2018). So while KROPE *representations* stabilize value function learning, KROPE's *learning algorithm* may diverge and not converge to KROPE representations. However, regardless of this result, KROPE does improve the stability and accuracy of FQE in all cases.

C.3.3 Stability-related Analysis on Custom Datasets

In this section, We include the remaining stability-related metric analysis that was deferred from the main paper.

Feature Co-adaptation, Condition Number, and Hyperparameter Sensitivity. In this subsection, we include all the remaining results related to the stability metrics for all environments.

Bellman completeness. Another metric that is associated with stability is Bellman completeness (BC) (Chang et al., 2022; Wang et al., 2021a). We find that KROPE is approximately Bellman complete even though it does not explicitly optimize for it; this finding aligns with our Theorem 2. While BC is difficult to approximate, we can minimize the proxy metric introduced given in Equation (10) (Chang et al., 2022):

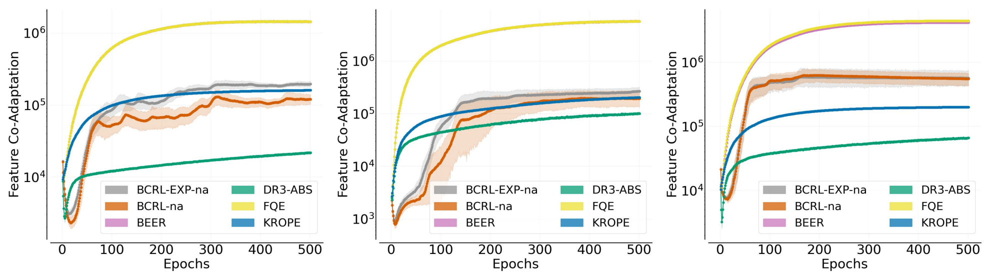


(a) Cheetah random (b) Cheetah medium (c) Cheetah medium-expert

(d) Hopper random (e) Hopper medium (f) Hopper medium-expert

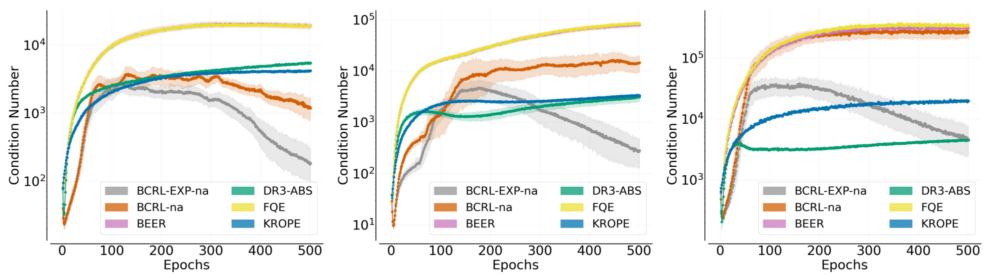
(g) Walker random (h) Walker medium (i) Walker medium-expert

Figure 6: Normalized squared value error achieved by LSPE when using a particular representation vs. representation training epochs on the D4RL datasets. LSPE estimates are computed every 10 epochs. Results are averaged over 20 trials and the shaded region is the 95% confidence interval. Lower and less erratic is better.



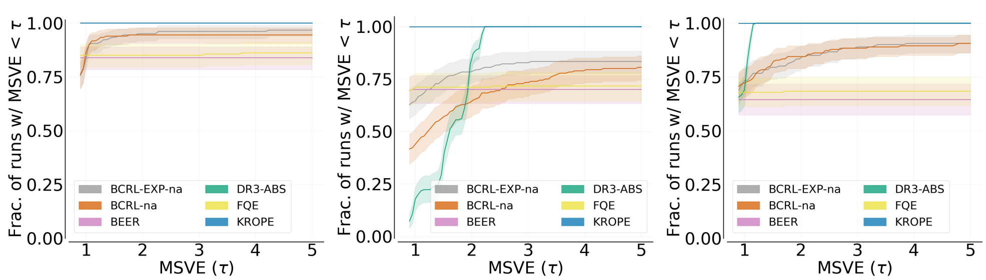
(a) CheetahRun (b) FingerEasy (c) WalkerStand

Figure 7: Feature co-adaptation on different environments as a function of training epochs. All results are averaged over 20 trials and shaded region is the 95% confidence interval.



(a) CheetahRun (b) FingerEasy (c) WalkerStand

Figure 8: Condition number of the covariance matrix on different environments as a function of training epochs. All results are averaged over 20 trials and shaded region is the 95% confidence interval.



(a) CheetahRun (b) FingerEasy (c) WalkerStand

Figure 9: Hyperparameter sensitivity on different environments as a function of training epochs; larger area under the curve is better. All results are averaged over 20 trials for each hyperparameter configuration and shaded region is the 95% confidence interval.

$$\mathcal{L}(M, \rho) := \mathbb{E}_{\mathcal{D}} \left\| \begin{bmatrix} M \\ \rho^\top \end{bmatrix} \phi(s, a) - \begin{bmatrix} \gamma \mathbb{E}_{s' \sim P(\cdot|s,a), a' \sim \pi_e(\cdot|s')}[\phi(s', a')] \\ r(s, a) \end{bmatrix} \right\|_2^2 \tag{10}$$

where $(\rho, M) \in \mathbb{R}^{d \times d}$, ϕ is fixed, and $\mathcal{L}(M, \rho) = 0$ indicates Bellman completeness. Given the final learned representation, we compute and report the BC error in Table 2. We find that KROPE is approximately Bellman complete even though it does not explicitly optimize for it; this finding aligns with our Theorem 2. We note that BCRL-EXP is less Bellman complete since it also includes the exploratory objective in its loss function, which if maximized can reduce the Bellman completeness. While BCRL is more BC than BCRL-EXP, we found that it is less BC in general. We attribute this finding due to the difficulty in explicitly optimizing the BCRL objective which involves multiple neural networks (M, ρ, ϕ) and multiple loss functions on different scales (reward, self-prediction, log determinant regularization losses). KROPE can achieve approximate Bellman completeness without these optimization-related difficulties.

	Algorithm					
Domain	BCRL + EXP	BCRL	BEER	DR3	FQE	KROPE (ours)
CartPoleSwingUp	0.4 ± 0.1	0.2 ± 0.1	0.1 ± 0.0	0.0 ± 0.0	0.1 ± 0.0	0.0 ± 0.0
CheetahRun	3.3 ± 0.6	2.4 ± 0.5	0.7 ± 0.0	0.0 ± 0.0	0.7 ± 0.0	0.2 ± 0.0
FingerEasy	1.3 ± 0.6	0.7 ± 0.2	0.9 ± 0.0	137.0 ± 4.4	0.9 ± 0.0	0.2 ± 0.0
WalkerStand	10.4 ± 2.0	0.3 ± 0.1	0.5 ± 0.1	66.1 ± 0.6	0.6 ± 0.0	0.1 ± 0.0

Table 2: Bellman completeness measure for all algorithms across all domains. Results are averaged across 20 trials and the deviation shown is the 95% confidence interval. Values are rounded to the nearest single decimal.

C.3.4 USING FQE DIRECTLY FOR OPE

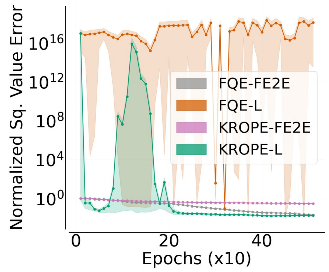
In our main empirical section (Section 4), we used FQE as a representation learning algorithm on our custom datasets. We adopted the linear evaluation protocol, i.e., an approach of analyzing the penultimate features of the action-value function network and applied LSPE on top of these features for OPE. This protocol enabled us to better understand the nature of the learned features.

For the sake of completeness, we present results of FQE as an OPE algorithm where the action-value network is directly used to estimate the performance of π_e. We present the results in Figures 10 and 11. As done in Section 4, we evaluate the performance of FQE and KROPE based on how they shape the penultimate features of the action-value network. However, when conducting OPE, we evaluate two variants: 1) using LSPE (-L) and 2) using the same end-to-end FQE action-value network (-E2E).

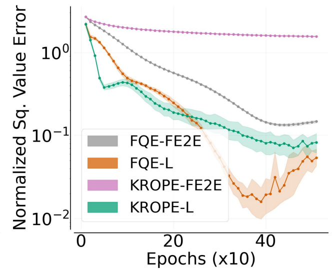
From Figure 10, we find that there are hyperparameter configurations that can outperform the KROPE variants. However, based on Figure 11, we find both KROPE variants are significantly more robust to hyperparameter tuning. This latter result suggests that KROPE does improve stability with respect to the hyperparameter sensitivity metric as well.

Regardless of FQE's hyperparameter sensitivity, it is still interesting to observe that when FQE is used as an OPE algorithm, it produces reasonably accurate OPE estimates. It even outperforms the FQE+KROPE combination. However, as shown in Section 4, the penultimate features of this same network actually have poor property values such as high feature co-adaptation, high condition numbers, and highly erratic OPE estimates with LSPE. Given that these features have weak properties, it is unclear why FQE still can lead to accurate OPE. The primary difference between using FQE for OPE vs. FQE features and LSPE for OPE is in how the last linear layer is trained. The former is trained by gradient descent while the latter is trained with the iterative LSPE algorithm on the fixed features. An interesting future direction will be to explore the learning dynamics of these two approaches.

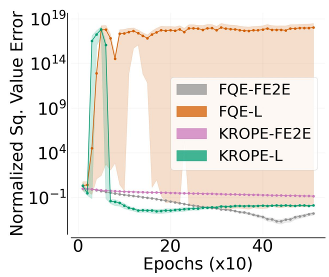
On a related note, we point out that the training dynamics of FQE are still not well-understood. For example, Fujimoto et al. (2022) show that the FQE loss function poorly correlates with value error. That is, the FQE loss can be high but value error (and OPE error) can be low.



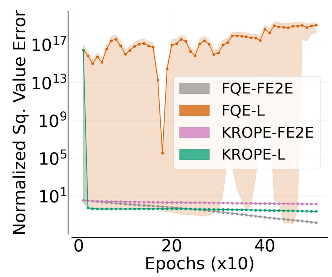
(a) CartPoleSwingUp



(b) FingerEasy

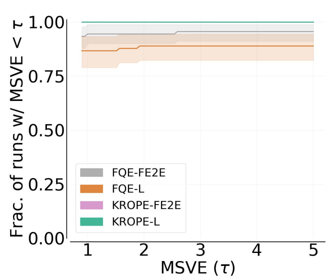


(c) CheetahRun

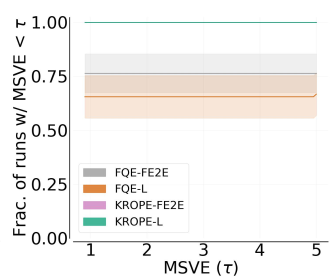


(d) WalkerStand

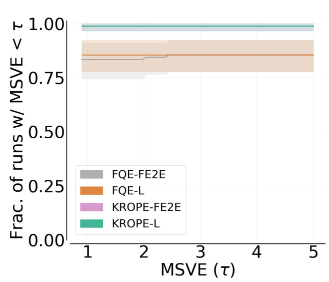
Figure 10: Normalized squared value error achieved by LSPE (-L) and FQE (-E2E) evaluated every 10 epochs of training. Results are averaged over 20 trials and the shaded region is the 95% confidence interval. Lower and less erratic is better.



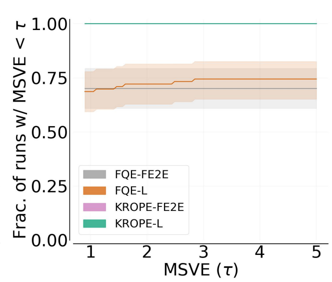
(a) CartPoleSwingUp



(b) FingerEasy



(c) CheetahRun



(d) WalkerStand

Figure 11: Hyperparameter sensitivity on different environments as a function of training epochs; larger area under the curve is better. All results are averaged over 10 trials for each hyperparameter configuration and shaded region is the 95% confidence interval. We tuned the hyperparameters discussed in Appendix C.1. KROPE-FE2E overlaps with KROPE-L.