

PIQL: Projective Implicit Q-Learning with Support Constraint for Offline Reinforcement Learning

Xinchen Han
SAMOVAR, Institut Polytechnique de
Paris
Palaiseau, France
xinchen.han@telecom-sudparis.eu

Hossam Afifi
SAMOVAR, Institut Polytechnique de
Paris
Palaiseau, France
hossam.afifi@telecom-sudparis.eu

Michel Marot
SAMOVAR, Institut Polytechnique de
Paris
Palaiseau, France
michel.marot@telecom-sudparis.eu

ABSTRACT

Offline Reinforcement Learning (RL) faces a fundamental challenge of extrapolation errors caused by out-of-distribution (OOD) actions. Implicit Q-Learning (IQL) employs expectile regression to achieve in-sample learning. Nevertheless, IQL relies on a fixed expectile hyperparameter and a density-based policy improvement method, both of which impede its adaptability and performance. In this paper, we propose Projective IQL (PIQL), a projective variant of IQL enhanced with a support constraint. In the policy evaluation stage, PIQL substitutes the fixed expectile hyperparameter with a projection-based parameter and extends the one-step value estimation to a multi-step formulation. In the policy improvement stage, PIQL adopts a support constraint instead of a density constraint, ensuring closer alignment with the policy evaluation. Theoretically, we demonstrate that PIQL maintains the expectile regression and in-sample learning framework, guarantees monotonic policy improvement, and introduces a progressively more rigorous criterion for advantageous actions. Experiments on D4RL and NeoRL2 benchmarks demonstrate robust gains across diverse domains, achieving state-of-the-art performance overall.¹

KEYWORDS

Offline RL; Support Constraint; Implicit Q-Learning; Vector Projection

ACM Reference Format:

Xinchen Han, Hossam Afifi, and Michel Marot. 2026. PIQL: Projective Implicit Q-Learning with Support Constraint for Offline Reinforcement Learning. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. <https://doi.org/10.65109/GZIN7614>

1 INTRODUCTION

Offline RL aims to learn policies from static datasets collected by unknown behavior policy, thereby avoiding costly or unsafe online interactions with the environment [20, 28]. This data-driven paradigm holds strong promise for real-world applications [14]. However, removing continuous interaction also introduces substantial challenges. A central difficulty is extrapolation error [9],

which arises from OOD actions that are not represented in the dataset. Such errors can be amplified through bootstrapped Bellman backups and may distort value estimates for a wide range of state-action pairs, leading to degraded policy performance. To address this challenge, existing research has elaborated offline RL algorithms focusing on policy improvement and policy evaluation, respectively.

In the policy improvement stage, policy constraint methods encourage the learned policy to remain close to the behavior policy. Behavior cloning (BC) [27] imitates dataset actions via supervised learning, while weighted BC (wBC) methods [12, 16, 23, 24, 26, 32] generalize BC by reweighting samples according to different criteria. Many wBC-style objectives can be interpreted as imposing a KL-style [5] regularization, which can be categorized into density constraint methods. In contrast, support constraint methods [18, 21, 22, 33] apply looser constraints that encourage the learned policy to remain within the support of the behavior policy while permitting more substantial improvement.

In the policy evaluation stage, conservative value learning methods mitigate overestimation issues induced by OOD actions [15, 17, 19, 25]. Especially, IQL [17] employs expectile regression to approximate an upper expectile of the value distribution instead of relying on the traditional maximum operator in Q-Learning. However, IQL has two notable limitations. First, IQL depends on a fixed expectile hyperparameter, which often requires dataset-specific tuning to achieve good performance. Second, IQL performs essentially one-step method [1], limiting its extensibility to more powerful multi-step policy improvement framework.

To overcome these challenges, we propose Projective IQL (PIQL), a projective variant of IQL enhanced with a support constraint. PIQL introduces two crucial innovations. First, PIQL replaces the hyperparameter τ with projection-based adaptive parameter $\tau_{\text{proj}}(a|s)$. This parameter $\tau_{\text{proj}}(a|s)$ is computed by projecting the behavior policy onto the learned policy, thereby eliminating the need for dataset-specific tuning. This formulation preserves the in-sample learning and expectile regression framework of IQL, while naturally generalizing one-step value estimation to a multi-step formulation. Second, PIQL incorporates support constraint aligned with the policy evaluation stage, achieving more effective policy improvement while mitigating extrapolation error.

Theoretically, we demonstrate that PIQL can be interpreted as a multi-step offline RL method while retaining the expectile-regression objective and the in-sample learning paradigm. Moreover, PIQL is provably guaranteed to yield monotonic policy improvement and to enforce a progressively more rigorous criterion for identifying and selecting advantageous actions.

¹Full version on <https://arxiv.org/abs/2501.08907>



This work is licensed under a Creative Commons Attribution International 4.0 License.

Experimentally, PIQL achieves state-of-the-art performance on D4RL [7], with particularly strong results on challenging “stitching” navigation tasks that require composing sub-trajectories over long horizons. We further evaluate PIQL on the recent NeoRL2 [10] benchmarks, where it consistently surpasses the strongest baselines.

2 RELATED WORK

To more effectively compare the advantages of PIQL, we broadly classify existing offline RL approaches into two categories: conservative value methods and policy constraint methods.

Conservative value methods aim to mitigate the overestimation of OOD actions by penalizing overly optimistic value estimates. CQL [19] introduces penalties on the Q -function for OOD actions, ensuring policy performance while staying close to the behavior policy. Extensions like ACL-QL [34] dynamically adjusts penalty term to reduce over-conservatism, and CSVE [3] penalizes state value function estimates across the dataset to enhance generalization. Despite their efficacy in avoiding overestimation, these approaches still face OOD action risk. In-sample learning approaches such as Onestep [1], IQL [17], and IVR [36] eliminate the need to estimate Q -values for OOD actions entirely. Onestep demonstrates strong results with one-step policy improvement but relies on well-covered datasets. IVR introduces an implicit value regularization framework, but Sparse Q-Learning within IVR relies on a restrictive assumption to simplify its optimization objective. IQL leverages the expectile regression framework for value learning, but its hyperparameter τ requires careful tuning, which is time-consuming.

Policy constraint methods enforce closeness between the learned policy and the behavior policy using various metrics such as KL divergence [23, 37], regularization terms [8], or CVAE model [4, 9, 13, 38], etc. While these density-based methods perform well under moderate suboptimality, they struggle when the dataset predominantly consists of poor-quality data. Support-based approaches relax density constraints to allow greater optimization flexibility. BEAR [18] keep the policy within the support of behavior policy by Maximum Mean Discrepancy (MMD). By contrast, SPOT [33] adopts a VAE-based density estimator to explicitly model the support set of behavior policy. Similarly, DASCO [30] employs discriminator scores in Generative Adversarial Networks (GANs). Although these methods achieve strong empirical performance, they lack rigorous policy improvement guarantees [21].

In contrast to conservative value methods, PIQL preserves the in-sample learning benefits of IQL while extending it into a multi-step formulation through a projection-based parameter. Unlike density-based policy constraint approaches, PIQL belongs to the family of support-constraint methods, thereby enabling greater policy optimization flexibility. Moreover, although PIQL adopts a policy improvement scheme similar to STR [21], PIQL further enforces increasingly rigorous criteria for selecting advantageous actions from the dataset, thereby ensuring steady policy improvement.

3 PRELIMINARIES

3.1 Offline RL

Conventionally, the RL problem is defined as a Markov Decision Process (MDP) [29], specified by a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{T}, r, \gamma \rangle$, where \mathcal{S} denotes the state space, \mathcal{A} is the action space, \mathcal{T} represents the

transition function, r is the reward function, and $\gamma \in [0, 1)$ denotes the discount factor.

For policy $\pi(a|s)$, the Q -function $Q^\pi(s, a)$ is defined as the expectation of cumulative discounted future rewards,

$$Q^\pi(s, a) = \mathbb{E}_{\substack{a_t \sim \pi(\cdot|s_t), \\ s_{t+1} \sim \mathcal{T}(\cdot|s_t, a_t)}} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \middle| s_0 = s, a_0 = a \right]. \quad (1)$$

The value function $V^\pi(s) = \mathbb{E}_{a \sim \pi} [Q^\pi(s, a)]$, and the advantage function $A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$. The Q -function is updated by minimizing the Temporal Difference (TD) error, shown in the following Eq. (2),

$$L_{\text{TD}}(\theta) = \mathbb{E}_{(s, a, s') \sim \mathcal{B}} \left[[r(s, a) + \gamma \max_{a'} Q_\theta(s', a') - Q_\theta(s, a)]^2 \right]. \quad (2)$$

When the state-action space is large, the Q -function is approximated by neural networks $Q_\theta(s, a)$ and parameterized by θ , and the $Q_\theta(s, a)$ represents the target- Q networks whose parameters $\hat{\theta}$ are updated via Polyak averaging approach.

In the offline RL setting, the replay buffer \mathcal{B} is replaced by the static dataset \mathcal{D} , and the agent is trained solely on \mathcal{D} without any interaction with the environment for further data collection. If an OOD action $a' = \arg \max_{a'} Q(s', a')$ is chosen but is absent from the dataset $a' \notin \mathcal{D}$, the estimated value $Q(s', a')$ becomes arbitrary and unreliable. Such pathological estimates can destabilize the bootstrapping process.

3.2 Support Constraint

The support-constrained policy class Π is defined as

$$\Pi = \{ \pi | \pi(a|s) = 0 \text{ whenever } \pi_\beta(a|s) = 0 \}, \quad (3)$$

where π_β is the behavior policy used to build datasets.

The policy optimization objective of wBC methods are summarized in [21], shown as follows,

$$J_\pi^{\text{wBC}}(\phi) = \max_{\phi} \mathbb{E}_{\substack{s \sim \mathcal{D}, \\ a \sim \pi_{\text{base}}}} [f(s, a; \pi_{\text{pe}}) \log \pi_\phi(a|s)], \quad (4)$$

where π_{base} denotes the sampling policy, and π_{pe} denotes the evaluation policy. $f(s, a; \pi_{\text{pe}}) = \frac{1}{Z(s)} \exp\left(\frac{A^{\pi_{\text{pe}}}(s, a)}{\lambda}\right)$, λ is a temperature parameter and $Z(s) = \int_a \pi_{\text{base}}(a|s) \exp\left(\frac{A^{\pi_{\text{pe}}}(s, a)}{\lambda}\right) da$. But $Z(s)$ is usually omitted during training, because errors in the estimation of $Z(s)$ caused more harm than the benefit the method derived from estimating this value, thereby $f(s, a; \pi_{\text{pe}}) = \exp\left(\frac{A^{\pi_{\text{pe}}}(s, a)}{\lambda}\right)$.

Since $f(s, a; \pi_{\text{pe}}) > 0$, the policy update of wBC is an equal-support update, i.e. if we initialize π_{base} as π_β and choose the current policy π_ϕ to be π_{base} , by recursive improvement, π_ϕ is still within the support of π_β . Based on the key observation above, STR [21] proposes a support constraint policy improvement method with $\pi_{\text{base}} = \pi_\phi$, $\pi_{\text{pe}} = \pi_\phi$ and adopts the Importance Sampling (IS) technique,

$$J_\pi^{\text{STR}}(\phi) = \max_{\phi} \mathbb{E}_{(s, a) \sim \mathcal{D}} \left[\frac{\bar{\pi}_\phi(a|s)}{\pi_\beta(a|s)} f(s, a; \bar{\pi}_\phi) \log \pi_\phi(a|s) \right], \quad (5)$$

where $\bar{\pi}_\phi(a|s)$ is the learned policy with detached of gradient.

3.3 IQL

IQL utilizes expectile regression to modify the policy evaluation objective in Eq. (2). The maximum operator is replaced by an upper expectile. In addition, to mitigate the stochasticity introduced by environment dynamics $s' \sim p(\cdot|s, a)$, IQL employs a separate value function that estimates an expectile solely with respect to the action distribution. Hence, the update of value function and the Q -function in IQL during policy evaluation are illustrated below:

$$L_V^{\text{IQL}}(\psi) = \mathbb{E}_{(s,a) \sim \mathcal{D}} [L_2^\tau(Q_\theta(s, a) - V_\psi(s))], \quad (6)$$

$$L_Q^{\text{IQL}}(\theta) = \mathbb{E}_{(s,a,s') \sim \mathcal{D}} [(r(s, a) + \gamma V_\psi(s') - Q_\theta(s, a))^2], \quad (7)$$

where $L_2^\tau(u) = |\tau - \mathbb{I}(u < 0)|u^2$, the τ expectile of some random variable is defined as a solution to the asymmetric least squares problem, and $\tau \in (0, 1)$.

IQL extract policy using the wBC method. The objective of policy improvement in IQL is defined as follows,

$$J_\pi^{\text{IQL}}(\phi) = \max_{\phi} \mathbb{E}_{(s,a) \sim \mathcal{D}} [f(s, a; \pi_\beta) \log \pi_\phi(a|s)]. \quad (8)$$

Following the definition of one-step algorithms [1], IQL can be classified as a one-step algorithm since the actions used for policy evaluation are drawn directly from the behavior policy, i.e., $a \sim \pi_\beta$ in $f(s, a; \pi_\beta)$. Furthermore, the actions for policy improvement are also restricted to the dataset $a \sim \mathcal{D}$, which allows IQL to benefit from in-sample learning and thus avoid OOD risk. However, this property simultaneously limits its potential for policy improvement.

4 PIQL

In this section, we provide a detailed description of PIQL, including policy evaluation, policy improvement, and practical implementation.

4.1 Policy Evaluation in PIQL

For the sake of simplicity, we follow the notation in IQL [17] to introduce PIQL. Let $\mathbb{E}_{X \sim \mathcal{X}}^\tau[x]$ be a τ^{th} expectile of X , e.g. $\mathbb{E}^{0.5}$ is the standard expectation. Then the recursive relationship between $V_\tau(s)$ and $Q_\tau(s, a)$ are shown as follows,

$$V_\tau(s) = \mathbb{E}_a^\tau [Q_\tau(s, a)]. \quad (9)$$

$$Q_\tau(s, a) = r(s, a) + \gamma \mathbb{E}_{s' \sim \mathcal{T}(\cdot|s,a)} [V_\tau(s')]. \quad (10)$$

Lemma 1. For all s , τ_1 and τ_2 such that $\tau_1 < \tau_2$ we get

$$V_{\tau_1}(s) \leq V_{\tau_2}(s).$$

Based on Lemma 1, a larger τ leads to a more optimistic policy evaluation, with the Q -function update approaching the maximum operator in Q -learning as $\tau \rightarrow 1$. Conversely, a smaller τ yields a more conservative policy evaluation. This observation highlights the role of τ as a trade-off parameter that balances optimism and conservatism in policy evaluation, and thus has a critical influence on the overall performance of IQL.

To illustrate the significance of the parameter τ , we evaluate the performance of IQL under three settings, $\tau \in \{0.3, 0.6, 0.9\}$, across 10 D4RL datasets. The results, presented in Fig. 1, show that the performance of IQL varies considerably with different values of τ

on the same dataset. This indicates that arbitrarily choosing τ may lead to suboptimal outcomes.

Nevertheless, what factors should be considered when determining τ ? To investigate this question, we revisit the wBC-based policy improvement method employed in IQL, as presented in Eq. (8). Formally, the objective of policy improvement can be interpreted as solving the following two-stage optimization problem, First stage:

$$\begin{aligned} \pi^*(a|s) &= \arg \max_{\pi} \mathbb{E}_{a \sim \pi(\cdot|s)} [A^{\pi_\beta}(s, a)], \\ \text{s.t. } D_{KL} [\pi(\cdot|s) || \pi_\beta(\cdot|s)] &\leq \epsilon, \\ \int_a \pi(a|s) da &= 1. \end{aligned} \quad (11)$$

Second stage:

$$\pi(\phi) = \arg \min_{\phi} \mathbb{E}_{\rho_{\pi_\beta}(s)} [D_{KL} [\pi^*(\cdot|s) || \pi_\phi(\cdot|s)]], \quad (12)$$

where π^* denotes the closed-form optimal solution, π_ϕ is the parametric policy to approximate π^* by minimizing the KL divergence, and $\rho_{\pi_\beta}(s)$ signifies the state distribution in dataset.

According to the constraints in optimization problem (11) and (12), the parameter τ should first be designed to reflect the distance $D(\pi_\phi || \pi_\beta)$ between the learned policy π_ϕ and the behavior policy π_β . As π_ϕ approaches π_β , τ should increase optimistically; conversely, τ should decrease conservatively as they diverge.

Vector projection provides a way to identify the component of a vector along the direction of another vector. Specifically, the vector projection of vector \vec{u} onto \vec{v} is defined as follows,

$$\text{Proj}_{\vec{v}} \vec{u} = \frac{\vec{u} \cdot \vec{v}}{\|\vec{v}\|^2} \vec{v} = \frac{\vec{u} \cdot \vec{v}}{\|\vec{v}\|} \frac{\vec{v}}{\|\vec{v}\|},$$

where $\|\vec{x}\| = \sqrt{\sum_{i=1}^m x_i^2}$ denotes the l_2 -norm of the vector $\vec{x} \in \mathbb{R}^m$.

In practice, the learned policy π_ϕ is approximated by neural networks and represented by a K -dimensional diagonal Gaussian distribution, where K represents the dimension of the action space. Typically, the policy networks are trained and updated using batch samples. Given n input samples $[(s_1, a_1), (s_2, a_2), \dots, (s_n, a_n)]$, the policy network outputs $\vec{\pi}_\phi(a|s) = [\pi_\phi(a_1|s_1), \pi_\phi(a_2|s_2), \dots, \pi_\phi(a_n|s_n)]$. Similarly, the behavior policy π_β can also be approximated by a neural network, and output $\vec{\pi}_\beta(a|s) = [\pi_\beta(a_1|s_1), \pi_\beta(a_2|s_2), \dots, \pi_\beta(a_n|s_n)]$.

Therefore, we propose parameter $\vec{\tau}_{\text{proj}}(a|s)$ based on the vector projection $\text{Proj}_{\vec{\pi}_\phi} \vec{\pi}_\beta$ of behavior policy vector $\vec{\pi}_\beta(a|s)$ onto the learned policy vector $\vec{\pi}_\phi(a|s)$,

$$\vec{\tau}_{\text{proj}}(a|s) = \frac{\vec{\pi}_\beta(a|s) \cdot \vec{\pi}_\phi(a|s)}{\|\vec{\pi}_\phi(a|s)\|^2} \vec{\pi}_\phi(a|s). \quad (13)$$

For one state-action pair in a batch sample, i.e. each component $\tau_{\text{proj}}(a|s)$ of the vector $\vec{\tau}_{\text{proj}}(a|s)$ is defined as follows,

$$\tau_{\text{proj}}(a|s) = \frac{\vec{\pi}_\beta(a|s) \cdot \vec{\pi}_\phi(a|s)}{\|\vec{\pi}_\phi(a|s)\|^2} \pi_\phi(a|s). \quad (14)$$

Based on the $\tau_{\text{proj}}(a|s)$, the value function and Q -function for policy evaluation in PIQL are optimized according to the following loss functions,

$$L_V(\psi) = \mathbb{E}_{(s,a) \sim \mathcal{D}} [L_2^{\tau_{\text{proj}}}(Q_\theta(s, a) - V_\psi(s))], \quad (15)$$

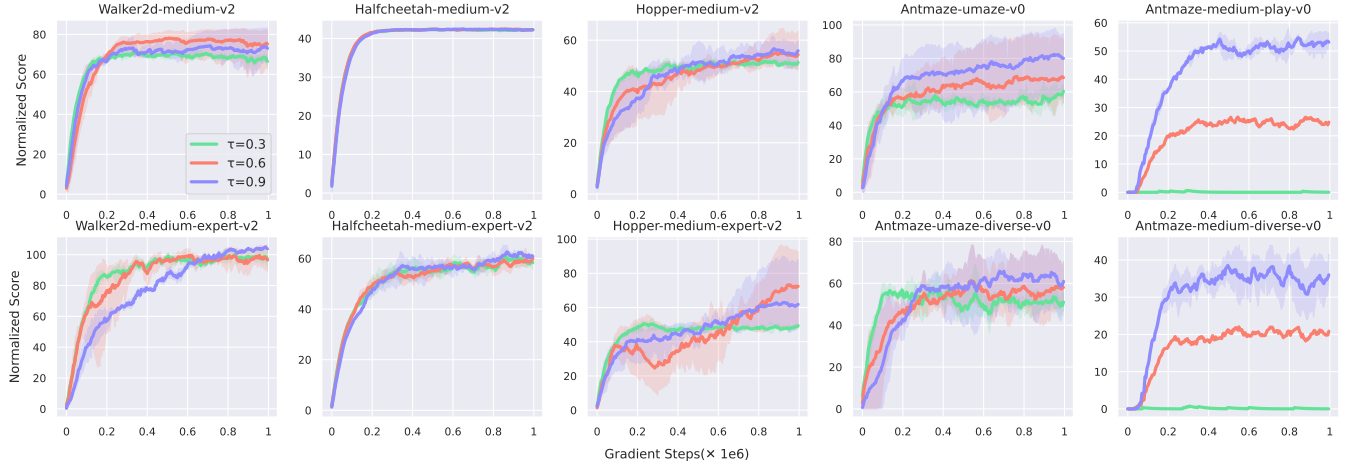


Figure 1: Effect of the Parameter $\tau \in \{0.3, 0.6, 0.9\}$ on IQL Performance across 10 Datasets.

$$L_Q(\theta) = \mathbb{E}_{(s,a,s') \sim \mathcal{D}} [(r(s,a) + \gamma V_\psi(s') - Q_\theta(s,a))^2]. \quad (16)$$

The advantages of $\tau_{\text{proj}}(a|s)$ for policy evaluation can be summarized as follows:

1. *Reflecting Distance and Adaptively Adjust.*

$\tau_{\text{proj}}(a|s)$ represents the length of the shadow of $\tilde{\pi}_\beta(a|s)$ over $\tilde{\pi}_\phi(a|s)$ at state-action pair (s, a) . As $\tilde{\pi}_\phi(a|s)$ gets closer to $\tilde{\pi}_\beta(a|s)$, the larger the projection, the larger $\tau_{\text{proj}}(a|s)$, thus allowing for an optimistic evaluation toward the policy $\pi_\phi(a|s)$; conversely, the small $\tau_{\text{proj}}(a|s)$ allows for a conservative evaluation. The trend of $\tau_{\text{proj}}(a|s)$ is consistent with the optimization problem (11) and (12). Moreover, $\tau_{\text{proj}}(a|s)$ is calculated based on the batch sample under the current policy, not a pre-set hyperparameter. Hence, $\tau_{\text{proj}}(a|s)$ can also adaptively adjust without the requirement for time-consuming fine-tuning.

2. *Multi-step yet In-sample Learning Paradigm.*

Based on the objective in Eq. (8), the actions for policy evaluation are drawn from the behavior policy $a \sim \pi_\beta$ in $f(s; a; \pi_\beta)$, and the actions for policy improvement are from the dataset $a \sim \mathcal{D}$. Therefore, IQL is essentially a one-step algorithm. By contrast, PIQL can be classified as a multi-step algorithm, as the τ_{proj} parameter of value function Eq. (15) is determined based on information from the policy π_ϕ . Furthermore, PIQL retains the expectile regression framework and the in-sample learning paradigm, effectively mitigating the risks associated with OOD actions. This property is formally established in Theorem 1.

Theorem 1. *If the ϵ in the constraint Eq. (11) is sufficiently small for $\{a|\pi_\beta(a|s) > 0, \pi_\phi(a|s) > 0, \text{ and } Q_\delta(s, a) - V_\psi(s) < 0, \text{ for all } s\}$, Eq. (15) will be transformed as follows,*

$$L_V(\psi) = \mathbb{E}_{\substack{s \sim \mathcal{D}, \\ a \sim \pi_\phi(\cdot|s)}} \left[L_2^{\bar{\tau}_{\text{proj}}} (Q_\delta(s, a) - V_\psi(s)) \right],$$

where $\bar{\tau}_{\text{proj}}(a|s) = \frac{\tilde{\pi}_\beta(a|s) \cdot \tilde{\pi}_\phi(a|s)}{\|\tilde{\pi}_\phi(a|s)\|^2} \pi_\beta(a|s)$.

In Theorem 1, the loss function $L_V(\psi)$ is demonstrated to be equivalent to Eq. (15). This equivalence implies that in Eq. (15),

the actions $a \sim \mathcal{D}$, when combined with the projection parameter τ_{proj} , can be equivalently represented as $a \sim \pi_\phi$. Hence, PIQL can be interpreted as a multi-step algorithm. Furthermore, Theorem 1 can be reformulated within a complete expectile regression framework by transforming $\tau_{\text{proj}}(a|s)$ into $\bar{\tau}_{\text{proj}}(a|s)$. It should be emphasized that Theorem 1 is derived under certain assumptions: the ϵ is sufficiently small for actions $\{a|Q_\delta(s, a) - V_\psi(s) < 0\}$. These assumptions are reasonable, as actions $\{a|Q_\delta(s, a) - V_\psi(s) < 0\}$ are generally regarded as suboptimal. For such actions, the optimal policy is to adopt imitation to avoid OOD overestimation, rather than engaging in unnecessary exploration of suboptimal actions.

4.2 Policy Improvement in PIQL

The policy improvement method in IQL is the wBC method, shown in Eq. (8). Nevertheless, directly applying the wBC approach to PIQL does not yield satisfactory results. This is because the policy struggles to converge to the optimal solution when ϵ is sufficiently small. Further details and theoretical justification are illuminated in the Lemma 2.

Lemma 2. *If $D_{KL} [\pi_\phi(\cdot|s) || \pi_\beta(\cdot|s)] \leq \epsilon, \forall s$ is guaranteed, then the performance $\eta(\pi) = \mathbb{E}_{\tau \sim \pi} [\sum_{t=0}^{\infty} \gamma^t r_t]$ has the following bound*

$$\eta(\pi_\phi) \leq \eta(\pi_\beta) + \frac{V_{\max}}{\sqrt{2}(1-\gamma)} \sqrt{\epsilon},$$

where $0 \leq Q^\pi, V^\pi \leq \frac{R_{\max}}{1-\gamma} =: V_{\max}$.

Based on the Lemma 2, to achieve the optimal policy with high probability, ϵ must be set to a large value. However, this requirement contradicts the assumption in Theorem 1 of the policy evaluation stage, rendering the wBC policy improvement approach unsuitable for PIQL. By contrast, the support constraint-based policy improvement method, as presented in Eq. (5), relaxes the density constraint. Consequently, we adopt the support constraint-based method for

policy improvement in PIQL, shown as follows,

$$\begin{aligned}
J_\pi(\phi) &= \max_{\phi} \mathbb{E}_{\substack{s \sim \mathcal{D}, \\ a \sim \pi_{\phi}(\cdot|s)}} \left[f(s, a; \bar{\pi}_{\phi}, \bar{\tau}_{\text{proj}}) \log \pi_{\phi}(a|s) \right] \\
&= \max_{\phi} \mathbb{E}_{(s,a) \sim \mathcal{D}} \left[\frac{\bar{\pi}_{\phi}(a|s)}{\pi_{\beta}(a|s)} f(s, a; \bar{\pi}_{\phi}, \bar{\tau}_{\text{proj}}) \log \pi_{\phi}(a|s) \right] \\
&= \max_{\phi} \mathbb{E}_{(s,a) \sim \mathcal{D}} \left[\frac{\bar{\pi}_{\phi}(a|s)}{\pi_{\beta}(a|s)} \exp\left(\frac{A_{\bar{\tau}_{\text{proj}}}^{\bar{\pi}_{\phi}}(s, a)}{\lambda}\right) \log \pi_{\phi}(a|s) \right],
\end{aligned} \tag{17}$$

where $A_{\bar{\tau}_{\text{proj}}}^{\bar{\pi}_{\phi}}(s, a) = Q_{\bar{\tau}_{\text{proj}}}^{\bar{\pi}_{\phi}}(s, a) - V_{\bar{\tau}_{\text{proj}}}^{\bar{\pi}_{\phi}}(s)$.

With an exact Q -function, we demonstrate that PIQL guarantees strict policy improvement, as demonstrated in Theorem 2.

Theorem 2. *If we have exact Q -function and $\tau_{k+1}(a|s) \geq \tau_k(a|s)$ ², then π_k in PIQL enjoys monotonic improvement:*

$$Q_{\tau_{k+1}}^{\pi_{k+1}}(s, a) \geq Q_{\tau_k}^{\pi_k}(s, a), \quad \forall s, a.$$

Theorem 2 establishes that PIQL ensures monotonic policy improvement when policy optimization is performed under support constraint.

Compared with STR [21], which also adopts a support-constraint approach, PIQL exhibits a clear advantage by enforcing a more rigorous criterion for evaluating *advantageous actions*, where advantageous actions are defined as $\{a|Q(s, a) - V(s) \geq 0\}$. Intuitively, progressively tightening the advantageous actions evaluation criterion leads to a more “concentrated” policy, i.e., one that selects advantageous actions with higher probability.

To illustrate this advantage, we utilize the Monte Carlo estimates approach. Specifically, we sample a large number of actions according to a certain policy $a \sim \pi(\cdot|s)$ and compare the probabilities associated with advantageous actions. If

$$P \{Q_{\tau_{k+1}}^{\pi_{k+1}}(s, a) - V_{\tau_{k+1}}^{\pi_{k+1}}(s) \geq 0\} \leq P \{Q_{\tau_k}^{\pi_k}(s, a) - V_{\tau_k}^{\pi_k}(s) \geq 0\}, \tag{18}$$

Eq. (18) holds, then we define the evaluation criterion for advantageous actions as being more rigorous at $k + 1$ -th iteration, where $P\{\mathcal{M}\}$ denotes the probability of event \mathcal{M} .

We demonstrate that the expectation form of this more rigorous evaluation criterion is satisfied in Theorem 3.

Theorem 3. *(Expectation Rigorous Criterion for Advantageous Actions). For $\forall s \in \mathcal{D}$ and $a \sim \pi_{k+1}(\cdot|s)$, if $0.5 \leq \tau_k(a|s), \tau_{k+1}(a|s) \leq 1$, we have*

$$\mathbb{E}_{a \sim \pi_{k+1}(\cdot|s)} \left[Q_{\tau_{k+1}}^{\pi_{k+1}}(s, a) - V_{\tau_{k+1}}^{\pi_{k+1}}(s) \right] \leq \mathbb{E}_{a \sim \pi_{k+1}(\cdot|s)} \left[Q_{\tau_k}^{\pi_k}(s, a) - V_{\tau_k}^{\pi_k}(s) \right].$$

Theorem 3 establishes the expectation form of the advantageous action evaluation criterion. Specifically, it states that the expectation of advantage function $A_{\tau_{k+1}}^{\pi_{k+1}}(s, a)$ under the policy π_{k+1} is lower than the expectation of $A_{\tau_k}^{\pi_k}(s, a)$. Intuitively, for the same sampling policy π_{k+1} , if $A_{\tau_{k+1}}^{\pi_{k+1}}(s, a)$ is smaller than $A_{\tau_k}^{\pi_k}(s, a)$, then the corresponding expectation is also smaller. This result indicates that the evaluation criterion for advantageous actions at the $(k + 1)$ -th iteration, $A_{\tau_{k+1}}^{\pi_{k+1}}(s, a)$, is more rigorous than the previous criterion $A_{\tau_k}^{\pi_k}(s, a)$, as formalized in Eq. (18).

²To simplify notation and without ambiguity, we denote the $\tau_{\text{proj}}(a|s)$ at the k -th iteration as $\tau_k(a|s)$.

Furthermore, Theorem 4 extends result in Theorem 3 to the maximum case of more rigorous evaluation criterion for two arbitrary policies. Before presenting Theorem 4, we first introduce Lemma 3 as follows,

Lemma 3. *For any random variable X , if $0.5 \leq \tau_1 \leq \tau_2 \leq 1$, we get*

$$\text{Var}^{\tau_1}(X) \leq \text{Var}^{\tau_2}(X),$$

where $\text{Var}^{\tau}(X) = \mathbb{E}[(X - \mathbb{E}^{\tau}(X))^2]$.

Theorem 4. *(Maximum Rigorous Criterion for Advantageous Actions). For $\forall s \in \mathcal{D}$, if $0.5 \leq \tau_k(a|s) \leq \tau_{k+1}(a|s) \leq 1$, we have*

$$\begin{aligned}
\max_{a \sim \pi_1(\cdot|s)} P \{Q_{\tau_{k+1}}^{\pi_{k+1}}(s, a) - V_{\tau_{k+1}}^{\pi_{k+1}}(s) \geq 0\} &\leq \\
\max_{a \sim \pi_2(\cdot|s)} P \{Q_{\tau_k}^{\pi_k}(s, a) - V_{\tau_k}^{\pi_k}(s) \geq 0\}, &
\end{aligned}$$

where $\pi_1(\cdot|s)$ and $\pi_2(\cdot|s)$ are arbitrary policies.

Theorem 4 generalizes Theorem 3 by extending the sampling policy π_{k+1} to two arbitrary policies. Intuitively, Theorem 4 shows that the maximum probability of selecting advantageous actions becomes progressively more rigorous across iterations, regardless of which policies are utilized for sampling. This result indicates that even under the most favorable conditions, the evaluation criterion tightens over time, thereby ensuring that the policy improvement process consistently focuses on truly advantageous actions. In addition, Theorem 4 is established under the assumption that $\tau_k(a|s) \leq \tau_{k+1}(a|s)$. In subsequent experiments, $\tau_{\text{proj}}(a|s)$ exhibits an increasing trend, which supports the validity of this assumption.

4.3 Practical Implementation in PIQL

Behavior Policy. The behavior policy $\pi_{\beta}(a|s)$ is approximated by a neural network trained using the behavior cloning method, as defined in Eq. (19),

$$\pi_{\beta}(a|s) = \underset{\pi}{\operatorname{argmax}} \mathbb{E}_{(s,a) \sim \mathcal{D}} [\log \pi(a|s)]. \tag{19}$$

Parameter Processing. In both Theorem 3 and Theorem 4, we assume that the projection parameter satisfies $0.5 \leq \tau_{\text{proj}}(a|s) \leq 1$. In practice, this constraint is enforced using the *clip* function, defined as

$$\tau_{\text{proj}}(a|s) = \text{clip}(\tau_{\text{proj}}(a|s), 0.5, 1),$$

where $\text{clip}(x, l, u) = \min(\max(x, l), u)$.

In implementation, we observed that computing $\tau_{\text{proj}}(a|s)$ for each individual state–action pair leads to results that are overly fine-grained, thereby reducing the stability of policy evaluation. To mitigate this issue, we instead compute the mean value of $\tau_{\text{proj}}(a|s)$ across batch samples, which significantly improves stability.

Importance Sampling. To mitigate high variance as well as training instability, we employ the Self-Normalized Importance Sampling (SNIS) method [21], which normalizes the IS ratios across the batch.

Overall Architecture. Integrating the policy evaluation, policy improvement, and practical implementation details, the complete PIQL algorithm is summarized in Algorithm 1.

Table 1: Averaged normalized scores on Gym-MuJoCo-v2 datasets. We indicate in bold that the scores are the top-2.

Dataset	DT	RvS	POR	CQL	LAPO	TD3+BC	Reinformer	CRR	IQL	PIQL
halfcheetah-med	42.6	41.6	48.8	44.4	46.0	42.8	42.9	47.1	47.4	48.7±0.45
hopper-med	67.6	60.2	78.6	86.6	51.6	99.5	81.6	38.1	66.3	68.2±13.90
walker2d-med	74.0	71.7	81.1	74.5	80.8	79.7	80.5	59.7	78.3	83.4±0.63
halfcheetah-med-rep	36.6	38.0	43.5	46.2	43.9	43.3	39.0	44.4	44.2	45.5±0.84
hopper-med-rep	82.7	73.5	98.9	48.6	37.6	31.4	83.3	25.5	94.7	95.7±10.63
walker2d-med-rep	66.6	60.6	76.6	32.6	52.3	25.2	72.8	27.0	73.9	89.5±2.68
halfcheetah-med-exp	86.8	92.2	94.7	62.4	86.1	97.9	92.0	85.2	86.7	94.2±1.04
hopper-med-exp	107.6	101.7	90.0	111.0	100.7	112.2	107.8	53.0	91.5	101.5±19.82
walker2d-med-exp	108.1	106.0	109.1	98.7	109.4	101.1	109.3	91.3	109.6	112.2±0.93
Gym-MuJoCo-v2 total	672.6	645.5	721.3	605.0	608.4	633.1	709.2	471.3	692.4	738.9±50.92

Algorithm 1 PIQL

Input: Dataset $\mathcal{D} = \{(s, a, s', r)\}$.
Parameter: Q -network θ , target Q -network $\hat{\theta}$, value network ψ , policy network ϕ .

- 1: // **Pre-train behavior policy for N_β steps.**
- 2: **for** $t = 1$ to N_β **do**
- 3: Optimize the behavior policy by Eq. (19).
- 4: **end for**
- 5: // **Policy Evaluation and Improvement.**
- 6: **for** each gradient step **do**
- 7: Random Batch Sample $(s, a, s', r) \sim \mathcal{D}$.
- 8: Calculate $\tau_{\text{proj}}(a|s)$ by Eq. (14).
- 9: Calculate batch mean value of $\tau_{\text{proj}}(a|s)$.
- 10: Optimize value network ψ by Eq. (15).
- 11: Optimize Q -network θ by Eq. (16).
- 12: Optimize policy network ϕ by Eq. (17).
- 13: Soft-update target Q -network $\theta' \leftarrow (1 - \tau)\theta' + \tau\theta$.
- 14: **end for**
- 15: **return** Policy π_ϕ .

5 EXPERIMENTS

5.1 Comparisons on D4RL Benchmarks

Gym-MuJoCo Locomotion Tasks are popular benchmarks used in prior offline RL works. Accordingly, we compare PIQL with several competitive and commonly adopted baselines, including Decision Transformer (DT) [2], RvS [6], POR [35], CQL, LAPO [4], TD3+BC [8], Reinformer [39], CRR [32] and IQL [17]. The results are reported in Tab. 1, where we average the mean return over 10 evaluation trajectories and 5 random seeds, and additionally report the standard deviation.

Although the performance of various offline RL algorithms on the Gym-MuJoCo-v2 datasets has largely saturated, PIQL continues to exhibit strong results, achieving top-2 performance on 6 out of 9 tasks and obtaining the highest overall score across all 9 datasets, thereby demonstrating the effectiveness of the proposed approach.

AntMaze and Kitchen Navigation tasks present significantly greater challenges, compared to the Gym-MuJoCo locomotion tasks. The main challenge is to learn policies for long-horizon planning

from datasets that do not contain optimal trajectories. We evaluate PIQL against several competitive methods, including BC, BCQ [9], BEAR, CQL, Onestep, TD3+BC, Diffusion-QL [31], \mathcal{X} -QL [11], and IQL, utilizing the same dataset versions as employed in IQL for a fair comparison. The results are presented in Tab. 2.

The Tab. 2 reveal that imitation learning methods such as BC and BCQ struggle to achieve satisfactory results. This limitation arises due to sparse rewards and a large amount of suboptimal trajectories, which place higher demands on algorithms for stable and robust Q -function and value function estimation. PIQL addresses these challenges by dynamically and adaptively tuning the τ_{proj} parameter to estimate the Q -function and value function according to the current policy. As a result, PIQL demonstrates exceptional performance on the AntMaze-v0 and Kitchen-v0 datasets, particularly excelling in AntMaze, where it achieves the highest scores on 5 out of the 6 datasets.

5.2 The Training Curves of τ_{proj}

In contrast to the fixed hyperparameter τ in IQL, $\tau_{\text{proj}}(a|s)$ can be flexibly and adaptively tuned. Therefore, we plot the $\tau_{\text{proj}}(a|s)$ curve and normalized score curve on AntMaze-v0 and Kitchen-v0 datasets. The results are shown in Fig. 2.

Based on Fig. 2, we observed that the curve $\tau_{\text{proj}}(a|s)$ exhibits an upward trend, reflecting an increase in the projected component of $\pi_\phi(a|s)$ within $\pi_\beta(a|s)$. This phenomenon arises because, in PIQL, the updated method of policy can be seen as a weighted behavior cloning as described in Eq. (17), where the weight factor

is given by $\frac{\tilde{\pi}_\phi(a|s)}{\pi_\beta(a|s)} \exp\left(\frac{A_{\text{proj}}^{\pi_\phi}(s,a)}{\lambda}\right) > 0$. However, this weight is essentially different from the weight in the aforementioned wBC method, since the support constraint is constructed via importance sampling. Furthermore, this upward trend supports our assumption $0.5 \leq \tau_k(a|s) \leq \tau_{k+1}(a|s) \leq 1$ in Theorem 2 and Theorem 4.

Another noteworthy observation is that the rise in normalized scores lags behind the $\tau_{\text{proj}}(a|s)$ parameter, indicating that updates to $\tau_{\text{proj}}(a|s)$ contribute to more accurate estimates of the Q -function and value function. Then, the more accurate estimates facilitate continuous policy improvement, resulting in higher normalized scores. However, this phenomenon is less apparent in the kitchen-complete-v0 dataset, since this dataset is relatively small with only

Table 2: Averaged normalized scores on AntMaze-v0, and Kitchen-v0 datasets. We indicate in bold that the scores are the top-2.

Dataset	BC	BCQ	BEAR	CQL	Onestep	TD3+BC	Diffusion-QL	X-QL	IQL	PIQL
antmaze-umaze	54.6	78.9	73.0	74.0	62.4	78.6	93.4	93.8	87.5	98.0±6.0
antmaze-umaze-diverse	45.6	55.0	61.0	84.0	44.7	71.4	66.2	82.0	62.2	98.0±6.0
antmaze-med-play	0.0	0.0	0.0	61.2	5.4	10.6	76.6	76.0	71.2	94.0±14.0
antmaze-med-diverse	0.0	0.0	8.0	53.7	1.8	3.0	78.6	73.6	70.0	88.0±8.0
antmaze-large-play	0.0	6.7	0.0	15.8	0.1	0.2	46.4	46.5	39.6	72.0±9.8
antmaze-large-diverse	0.0	2.2	0.0	14.9	0.9	0.0	56.6	49.0	47.5	50.0±10.0
AntMaze-v0 total	100.2	142.8	142.0	303.6	115.2	163.8	417.8	420.9	378.0	500.0±53.8
kitchen-complete	65.0	8.1	0.0	43.8	66.0	61.5	84.0	82.4	62.5	87.0±4.6
kitchen-partial	38.0	18.9	13.1	49.8	59.3	52.8	60.5	73.7	46.3	67.0±6.0
kitchen-mixed	51.5	8.1	47.2	51.0	56.5	60.8	62.6	62.5	51.0	55.5±3.2
Kitchen-v0 total	154.5	35.1	60.3	144.6	181.7	175.1	207.1	218.6	159.8	209.5±13.8

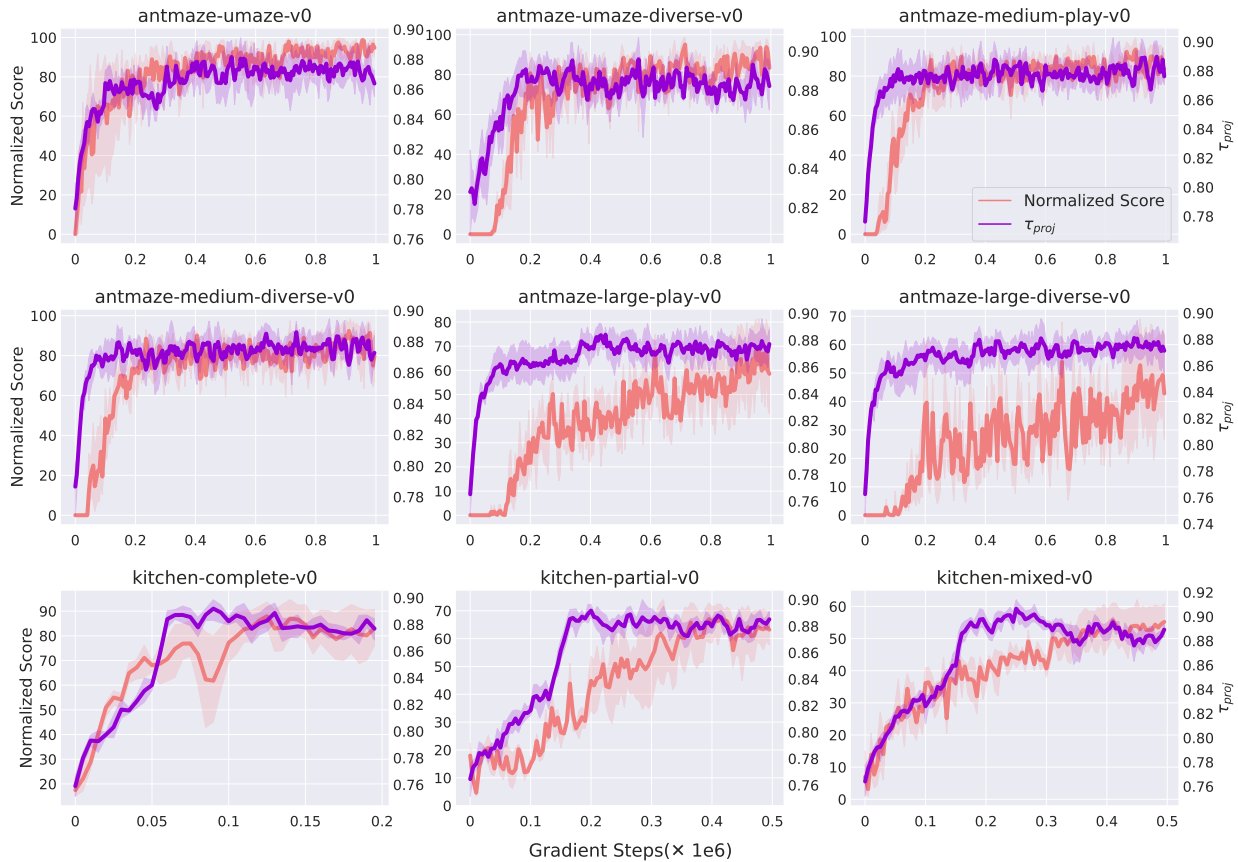


Figure 2: The training curves of normalized score and $\tau_{proj}(a|s)$ on AntMaze-v0 and Kitchen-v0 datasets.

3, 680 samples. In contrast, datasets such as AntMaze-v0 typically contain approximately 10^6 samples, and other Kitchen-v0 datasets include 136, 950 samples.

5.3 Ablation Experiment

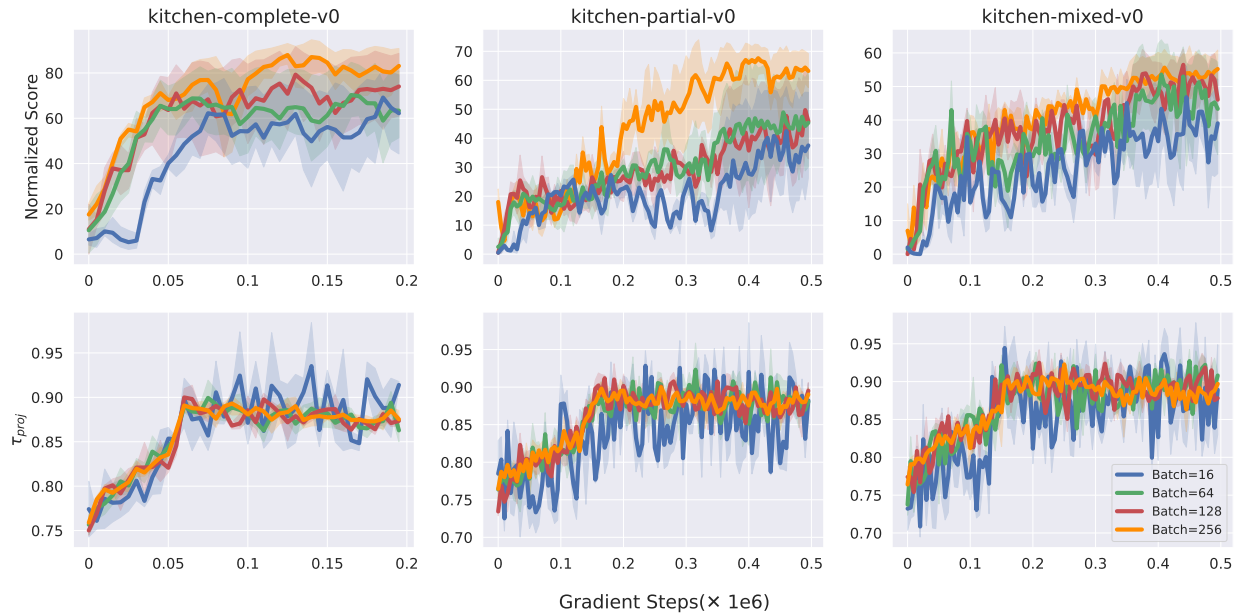
In the proposed PIQL method, each sample in the batch is treated as a component of the vector used to compute $\tau_{proj}(a|s)$, making the

batch size to be one of the critical parameters. To investigate this, we evaluated the performance of $\tau_{proj}(a|s)$ as well as normalized scores corresponding *batch size* = 16, 64, 128, 256 on three Kitchen-v0 datasets. The results are presented in Fig. 3.

As shown in Fig. 3, the normalized score gradually increases with larger batch sizes, which aligns with expectations. This improvement occurs because larger batch sizes allow both the policy

Table 3: Averaged normalized scores on NeoRL2 datasets. We indicate in bold that the scores are the top-2.

Dataset	Data	BC	CQL	EDAC	MCQ	TD3+BC	MOPO	COMBO	RAMBO	MOBILE	PIQL
Pipeline	69.3	68.6	81.1	72.9	49.7	82.0	-26.3	55.5	24.1	65.5	89.3±14.8
Simglucose	73.9	75.1	11.0	8.1	29.6	74.2	34.6	23.2	10.8	9.3	82.1±3.6
RocketRecovery	75.3	72.8	74.3	65.7	76.5	79.7	-27.7	74.7	-44.2	43.7	81.6±7.5
RandomFrictionHopper	28.7	28.0	33.0	34.7	31.7	29.5	32.5	34.1	29.6	35.1	34.2±0.7
DMSD	56.6	65.1	70.2	78.7	77.8	60.0	68.2	68.3	76.2	64.4	54.2±0.1
Fusion	48.8	55.2	55.9	58.0	49.7	54.6	-11.6	55.5	59.6	5.0	59.8±2.1
SafetyHalfCheetah	73.6	70.2	71.2	53.1	54.7	68.6	23.7	57.8	-422.4	8.7	70.7±1.2
NeoRL2 total	426.2	435.0	396.7	371.2	369.7	448.6	93.4	369.1	-266.3	231.7	471.9 ± 30.0

**Figure 3: The training curves of normalized score and $\tau_{\text{proj}}(a|s)$ on Kitchen-v0 datasets under different batch size.**

and value networks to be trained more effectively. However, this trend does not fully capture the specific impact of batch size on the parameter $\tau_{\text{proj}}(a|s)$. To address this issue, we also plotted the relationship between batch size and $\tau_{\text{proj}}(a|s)$. The results clearly indicate that the stability of $\tau_{\text{proj}}(a|s)$ improves as the batch size increases, with reduced fluctuations. A more stable $\tau_{\text{proj}}(a|s)$ is particularly beneficial for enhancing performance because of the policy evaluation is more stable.

5.4 Comparisons on NeoRL2 Benchmarks

NeoRL2 (Near Real-World Offline RL Benchmark) [10] is a recent benchmark designed to reflect practical constraints in offline RL, which includes domains motivated by industrial control, nuclear fusion, and healthcare. Moreover, practical data acquisition limits the dataset size and coverage.

We compare PIQL against baselines reported on NeoRL2, and the results are shown in Tab. 3. Based on Tab. 3, PIQL attains the best performance on 4 out of 7 datasets and yields the highest total score, substantially outperforming the strongest baseline.

6 CONCLUSION

We propose PIQL, an in-sample, multi-step, support-constraint offline RL algorithm based on IQL. During the policy evaluation phase, PIQL transitions from a one-step approach to a multi-step in-sample offline RL algorithm while preserving the expectile regression framework. In the policy improvement phase, PIQL incorporates a support constraint to facilitate effective policy updates. Furthermore, we provide a theoretical foundation, demonstrating the invariance of the expectile regression framework, monotonic policy improvement guarantees, and a more rigorous policy evaluation criterion. Empirical results validate the theoretical insights of PIQL and highlight its state-of-the-art performance on D4RL benchmarks. In the future, first, we will study how to make PIQL more data-efficient, aiming to learn strong policies from substantially smaller datasets. Second, we will explore replacing the behavior policy with more expressive models, aiming to further enhance robustness and generalization in practical applications.

REFERENCES

- [1] David Brandfonbrener, Will Whitney, Rajesh Ranganath, and Joan Bruna. 2021. Offline rl without off-policy evaluation. *Advances in neural information processing systems* 34 (2021), 4933–4946.
- [2] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Misha Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. 2021. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems* 34 (2021), 15084–15097.
- [3] Liting Chen, Jie Yan, Zhengdao Shao, Lu Wang, Qingwei Lin, Saravanakumar Rajmohan, Thomas Moscibroda, and Dongmei Zhang. 2024. Conservative state value estimation for offline reinforcement learning. *Advances in Neural Information Processing Systems* 36 (2024).
- [4] Xi Chen, Ali Ghadirzadeh, Tianhe Yu, Jianhao Wang, Alex Yuan Gao, Wenzhe Li, Liang Bin, Chelsea Finn, and Chongjie Zhang. 2022. Lapo: Latent-variable advantage-weighted policy optimization for offline reinforcement learning. *Advances in Neural Information Processing Systems* 35 (2022), 36902–36913.
- [5] Imre Csiszár. 1975. I-divergence geometry of probability distributions and minimization problems. *The annals of probability* (1975), 146–158.
- [6] Scott Emmons, Benjamin Eysenbach, Ilya Kostrikov, and Sergey Levine. 2021. Rvs: What is essential for offline rl via supervised learning? *arXiv preprint arXiv:2112.10751* (2021).
- [7] Justin Fu, Aviral Kumar, Ofri Nachum, George Tucker, and Sergey Levine. 2020. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219* (2020).
- [8] Scott Fujimoto and Shixiang Shane Gu. 2021. A minimalist approach to offline reinforcement learning. *Advances in neural information processing systems* 34 (2021), 20132–20145.
- [9] Scott Fujimoto, David Meger, and Doina Precup. 2019. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*. PMLR, 2052–2062.
- [10] Songyi Gao, Zuolin Tu, Rong-Jun Qin, Yi-Hao Sun, Xiong-Hui Chen, and Yang Yu. 2025. NeoRL-2: Near Real-World Benchmarks for Offline Reinforcement Learning with Extended Realistic Scenarios. *arXiv:2503.19267 [cs.LG]* <https://arxiv.org/abs/2503.19267>
- [11] Divyansh Garg, Joey Hejna, Matthieu Geist, and Stefano Ermon. 2023. Extreme q-learning: Maxent rl without entropy. *arXiv preprint arXiv:2301.02328* (2023).
- [12] Xinchun Han, Hossam Afifi, and Michel Marot. 2025. Cosine Similarity Based Adaptive Implicit Q-Learning for Offline Reinforcement Learning. In *2025 IEEE Wireless Communications and Networking Conference (WCNC)*. 1–6.
- [13] Xinchun Han, Hossam Afifi, and Michel Marot. 2025. ELAPSE: Expand Latent Action Projection Space for policy optimization in Offline Reinforcement Learning. *Neurocomputing* 631 (2025), 129665. <https://doi.org/10.1016/j.neucom.2025.129665>
- [14] Xinchun Han, Hossam Afifi, Hassine Mouncla, and Michel Marot. 2024. Leaky PPO: A Simple and Efficient RL Algorithm for Autonomous Vehicles. In *2024 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–7.
- [15] Xinchun Han, Hossam Afifi, Hassine Mouncla, and Michel Marot. 2025. Attention ensemble mixture: a novel offline reinforcement learning algorithm for autonomous vehicles. *Applied Intelligence* 55, 7 (March 2025), 14. <https://doi.org/10.1007/s10489-025-06403-7>
- [16] Xinchun Han, Qiuyang Fang, Hossam Afifi, and Michel Marot. 2026. Automatic Constraint Policy Optimization based on Continuous Constraint Interpolation Framework for Offline Reinforcement Learning. *arXiv:2601.23010 [cs.LG]* <https://arxiv.org/abs/2601.23010>
- [17] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. 2021. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169* (2021).
- [18] Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. 2019. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in neural information processing systems* 32 (2019).
- [19] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems* 33 (2020), 1179–1191.
- [20] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. 2020. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643* (2020).
- [21] Yixiu Mao, Hongchang Zhang, Chen Chen, Yi Xu, and Xiangyang Ji. 2023. Supported trust region optimization for offline reinforcement learning. In *International Conference on Machine Learning*. PMLR, 23829–23851.
- [22] Yixiu Mao, Hongchang Zhang, Chen Chen, Yi Xu, and Xiangyang Ji. 2024. Supported value regularization for offline reinforcement learning. *Advances in Neural Information Processing Systems* 36 (2024).
- [23] Ashvin Nair, Abhishek Gupta, Murtaza Dalal, and Sergey Levine. 2020. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359* (2020).
- [24] Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. 2019. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177* (2019).
- [25] Zhiyong Peng, Yadong Liu, Haoqiang Chen, and Zongtan Zhou. 2023. Conservative network for offline reinforcement learning. *Knowledge-Based Systems* 282 (2023), 111101.
- [26] Jan Peters and Stefan Schaal. 2007. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the 24th international conference on Machine learning*. 745–750.
- [27] Dean A Pomerleau. 1991. Efficient training of artificial neural networks for autonomous navigation. *Neural computation* 3, 1 (1991), 88–97.
- [28] Rafael Figueiredo Prudencio, Marcos ROA Maximo, and Esther Luna Colombini. 2023. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE Transactions on Neural Networks and Learning Systems* (2023).
- [29] Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.
- [30] Quan Vuong, Aviral Kumar, Sergey Levine, and Yevgen Chebotar. 2022. Dasco: Dual-generator adversarial support constrained offline reinforcement learning. *Advances in Neural Information Processing Systems* 35 (2022), 38937–38949.
- [31] Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. 2022. Diffusion policies as an expressive policy class for offline reinforcement learning. *arXiv preprint arXiv:2208.06193* (2022).
- [32] Ziyu Wang, Alexander Novikov, Konrad Zolna, Josh S Merel, Jost Tobias Springenberg, Scott E Reed, Bobak Shahriari, Noah Siegel, Caglar Gulcehre, Nicolas Heess, et al. 2020. Critic regularized regression. *Advances in Neural Information Processing Systems* 33 (2020), 7768–7778.
- [33] Jialong Wu, Haixu Wu, Zihan Qiu, Jianmin Wang, and Mingsheng Long. 2022. Supported policy optimization for offline reinforcement learning. *Advances in Neural Information Processing Systems* 35 (2022), 31278–31291.
- [34] Kun Wu, Yinuo Zhao, Zhiyuan Xu, Zhengping Che, Chengxiang Yin, Chi Harold Liu, Qinru Qiu, Feifei Feng, and Jian Tang. 2024. ACL-QL: Adaptive Conservative Level in Q-Learning for Offline Reinforcement Learning. *IEEE Transactions on Neural Networks and Learning Systems* (2024).
- [35] Haoran Xu, Li Jiang, Li Jianxiong, and Xianyuan Zhan. 2022. A policy-guided imitation approach for offline reinforcement learning. *Advances in Neural Information Processing Systems* 35 (2022), 4085–4098.
- [36] Haoran Xu, Li Jiang, Jianxiong Li, Zhuoran Yang, Zhaoran Wang, Victor Wai Kin Chan, and Xianyuan Zhan. 2023. Offline rl with no ood actions: In-sample learning via implicit value regularization. *arXiv preprint arXiv:2303.15810* (2023).
- [37] Chi Zhang, Sanmukh Kuppannagari, and Prasanna Viktor. 2021. Brac+: Improved behavior regularized actor critic for offline reinforcement learning. In *Asian Conference on Machine Learning*. PMLR, 204–219.
- [38] Wenxuan Zhou, Sujay Bajracharya, and David Held. 2021. Plas: Latent action space for offline reinforcement learning. In *Conference on Robot Learning*. PMLR, 1719–1735.
- [39] Zifeng Zhuang, Dengyun Peng, Jinxin Liu, Ziqi Zhang, and Donglin Wang. 2024. Reinformer: Max-return sequence modeling for offline rl. *arXiv preprint arXiv:2405.08740* (2024).