IN-CONTEXT COMPOSITIONAL Q-LEARNING FOR OFFLINE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Accurately estimating the Q-function is a central challenge in offline reinforcement learning. However, existing approaches often rely on a single global Q-function, which struggles to capture the compositional nature of tasks involving diverse subtasks. We propose In-context Compositional Q-Learning (`ICQL`), the first offline RL framework that formulates Q-learning as a contextual inference problem, using linear Transformers to adaptively infer local Q-functions from retrieved transitions without explicit subtask labels. Theoretically, we show that under two assumptions—linear approximability of the local Q-function and accurate weight inference from retrieved context—`ICQL` achieves bounded Q-function approximation error, and supports near-optimal policy extraction. Empirically, `ICQL` substantially improves performance in offline settings: improving performance in kitchen tasks by up to 16.4%, and in Gym and Adroit tasks by up to 8.6% and 6.3%. These results highlight the underexplored potential of in-context learning for robust and compositional value estimation, positioning `ICQL` as a principled and effective framework for offline RL.

1 INTRODUCTION

Offline reinforcement learning (Offline RL) aims to learn effective policies from fixed datasets without further interaction with the environment (Fujimoto et al., 2019; Lange et al., 2012). This setting is particularly important in real-world domains such as robotics (Kalashnikov et al., 2018), logistics (Wang et al., 2021), and operations research (Hubbs et al., 2020; Mazyavkina et al., 2021), where environment access is limited, data collection is expensive or risky, and historical data is often the only available resource. The central challenge of this modeling paradigm is the potential distributional shift: when the learned policy queries state-action pairs outside the dataset support, value function extrapolation can lead to severe overestimation and degenerate performance. (Fu et al., 2020; Kumar et al., 2020)

Contemporary methods primarily employ policy constraints (Chen et al., 2021b) or value regularization (Kumar et al., 2020; Kostrikov et al., 2021) to address this challenge. However, policy constraints are largely limited by the behavior policy that are used to collect offline data, and exhibit a trade-off between generalization and safe constraint adherence. While recent value regularization methods aim to provide conservative references for softer penalty on out-of-distribution actions, the optimality of the learned value function is not guaranteed due to limited and potentially biased static dataset.

We observe that, for each RL control task, the state space can be inherently divided into multiple sub-tasks. Although ideally a action-value function can be expressive enough to perfectly capture state-action value, the knowledge may not be fully transferrable among sub-tasks. For example, in Mujoco Locomotion tasks, knowledge about how to walk faster may not be helpful for solving how to recover from an unexpected non-nominal states. A visualization of this situation can be found in Figure 1, which shows the distribution of states after dimensionality reduction, colored by their actual future return in the offline dataset. Moreover, although states in the dataset can be grouped into coherent clusters, where each typically corresponding to a specific subtask, two clusters that appear geometrically may nevertheless correspond to semantically different behaviors and exhibit distinct long-horizon returns. Under the condition of insufficient offline data and inability of exploration, this property are not naturally captured by an offline value learning algorithm that fits a single global value function.

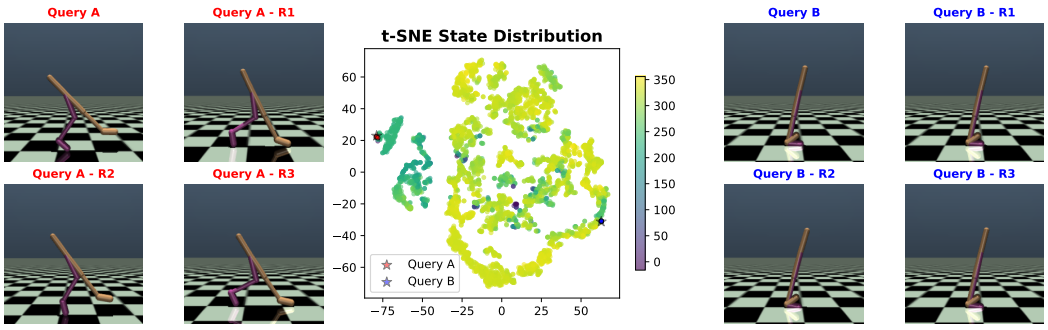


Figure 1: Center: dimension-reduced state distribution and corresponding value estimation by an SAC critic on `Walker2d-Medium-Expert` dataset. Left and right grids are two groups of similar states.

To address these challenges, we propose to cast value learning in offline reinforcement learning as a contextual inference problem, enabling local Q-function approximation via in-context learning. Specifically, we introduce In-context Compositional Q-Learning (ICQL), a general framework for offline RL that leverages the in-context learning capabilities of linear Transformers to infer local Q-functions from small, retrieved transition sets. Rather than fitting global approximators of value function, ICQL leverages the compositional nature and local structure of the task to learn the family of value functions, enabling flexible adaptation of value estimation locally within context windows. Our key contributions are summarized as follows:

- We introduce the first offline RL framework ICQL that **formulates Q-learning as a contextual inference problem**, leveraging in-context learning with linear Transformers to adaptively infer local Q-functions without requiring explicit subtask labels or structure.
- We provide a theoretical analysis showing that **ICQL achieves bounded approximation error** under two assumptions: linear approximability of the local Q-function and accurate weight inference from retrieved context, and prove the greedy policy with respect to it is guaranteed to be **near-optimal**.
- **ICQL improves the performance in offline settings through in-context local approximation**, and we demonstrate the effectiveness of our approach ICQL under both offline Q-learning and offline actor-critic frameworks. On the Gym and Adroit tasks, ICQL yields score improvements by **8.6%** and **6.3%**. Notably, on the Kitchen tasks, ICQL achieves a **16.4%** performance improvement over the second best baseline. We also show that ICQL does produce better value estimation. These results highlight the underexplored potential of linear attention in enabling robust and compositional value estimation for offline RL.
- We conduct extensive ablation studies to isolate the contributions of in-context learning and localized value inference. In addition, we investigate the impact of different retrieval strategies—including similarity metrics and context selection criteria—on overall performance and stability.

2 Related Work

Offline Reinforcement Learning. Offline RL aims to learn effective policies from static datasets without further environment interaction. Several recent approaches address distributional shift and overestimation in this setting by modifying Q-learning objectives or introducing conservative regularization. Notable examples include CQL (Kumar et al., 2020), IQL (Kostrikov et al., 2022) and TD3+BC (Fujimoto & Gu, 2021). CQL introduces a conservative penalty on Q-values for out-of-distribution actions to prevent value overestimation in offline settings. TD3+BC combines TD3 with behavior cloning loss to bias policy updates toward the dataset actions while retaining Q-learning. And IQL removes explicit policy optimization and learns value-weighted regression targets to implicitly extract high-value actions from offline data. These methods rely on global Q-function approximators trained across the entire state-action space, often leading to poor generalization

in compositional environments. In contrast, our approach decomposes value learning into local estimation problems, using in-context inference to adapt Q-functions to local transition dynamics without requiring additional supervision.

In-context Learning in RL. Recent work has applied Transformers to offline RL, using sequence modeling to learn return-conditioned policies (Zhao et al., 2025). For example, Decision Transformer (Chen et al., 2021a) and Gato (Reed et al., 2022) treat trajectories as sequences, while replay-based in-context RL (Chen et al., 2021a; Reed et al., 2022) uses Transformers for behavior cloning and reward learning. These approaches leverage the ability of pre-trained Transformers to adapt via prompt conditioning or in-context learning. In-context learning has shown both strong theoretical foundation (von Oswald et al., 2023; Shen et al., 2024; Wang et al., 2025b) and empirical performance across tasks (Hollmann et al., 2023; Micheli et al., 2023) and is increasingly studied in supervised settings (Laskin et al., 2023; Lee et al., 2023; Mukherjee et al., 2024). (Laskin et al., 2023) proposes Algorithm Distillation (`AD`) to mimic the data collection policy, but it is constrained by the quality of the original algorithm. `DPT` (Lee et al., 2023) improves regret in contextual bandits via in-context learning, but assumes access to optimal actions, which is often unrealistic in offline RL. `PreDeToR` (Mukherjee et al., 2024) adds reward prediction to decision transformers, yet still focuses on action generation. While these approaches focus on directly generating actions or policies from trajectories, they do not explicitly target value estimation, which are out of our research scope. Hence, we will not include these methods as our baselines. While recent works have explored Transformers in offline RL primarily for trajectory modeling or return-conditioned generation (Chen et al., 2021a; Laskin et al., 2023; Mukherjee et al., 2024) , we instead focus on using linear attention as a tool for in-context value learning. Our results suggest that linear attention, when applied for local Q-function estimation, offers strong performance and generalization benefits. To our knowledge, this is the first work to demonstrate such potential of linear attention for compositional value-based offline RL.

3 Methodology

3.1 Local Q-functions

In this section, we define the local Q-functions for offline RL based on the local neighborhood corresponding to each state. We define $\mathcal{D}$ as the dataset collecting all the offline transitions.

Definition 3.1. (Local Q-function Approximation) Given a transition $(s, a, r, s', a') \in \mathcal{D}$, there exist $d, \bar{d} > 0$ such that any nearby transition $(\bar{s}, \bar{a}, \bar{r}, \bar{s}', \bar{a}') \in \mathcal{D}$ is defined as

$$(\bar{s}, \bar{a}, \bar{r}, \bar{s}', \bar{a}') \in \left\{ (s_i, a_i, r_i, s_i', a_i') \in \mathcal{D} \middle| \|s_i - s\|_2^2 \leq d^2 \text{ and } \|s_i' - s_i\|_2^2 \leq \bar{d}^2 \right\} \triangleq \Omega_s^{(d,\bar{d})}. \quad (1)$$

For any transition $(\bar{s}, \bar{a}, \bar{r}, \bar{s}', \bar{a}') \in \Omega_s^{(d,\bar{d})}$, there exists an optimal uniform local weight vector w_s^* such that the local Q-function approximation is defined as

$$\hat{Q}_{\Omega_s^{(d,\bar{d})}}(\bar{s}, \bar{a}) \triangleq {w_s^*}^T \phi(\bar{s}, \bar{a}), \quad \forall (\bar{s}, \bar{a}, \bar{r}, \bar{s}', \bar{a}') \in \Omega_s^{(d,\bar{d})}, \quad (2)$$

where the function $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ is the feature function of the state-action pair $(\bar{s}, \bar{a})$. The best approximation of local Q-function $Q_{\Omega_s^{(d,\bar{d})}}(\bar{s}, \bar{a})$ is $\hat{Q}_{\Omega_s^{(d,\bar{d})}}(\bar{s}, \bar{a})$, i.e., there exists some $\varepsilon^s_{\text{approx}} > 0$ such that

$$\left| Q_{\Omega_s^{(d,\bar{d})}}(\bar{s}, \bar{a}) - {w_s^*}^\top \phi(\bar{s}, \bar{a}) \right| \leq \varepsilon^s_{\text{approx}}, \quad \forall (\bar{s}, \bar{a}, \bar{r}, \bar{s}', \bar{a}') \in \Omega_s^{(d,\bar{d})}. \quad (3)$$

In the rest of this paper, we will ignore $\bar{d}$ in the notation of $\Omega_s^{(d,\bar{d})}$ in Equation (1), since the condition $\|\bar{s}' - \bar{s}\|_2^2 \leq \bar{d}^2$ for some $\bar{d} > 0$ can be easily held in real continuous problems. We will use Ω_s^d to represent $\Omega_s^{(d,\bar{d})}$ instead. The local Q-function defined in Equation (2) is a local formalization for the general linear Q-function approximation, which has been widely used in previous research (Yin et al., 2022; Du et al., 2019; Poupart et al., 2002; Parr et al., 2008). We assume that for each local domain Ω_s^d, the local Q-function should have its own state-dependent local structure. This has been examined both theoretically and practically to give a better Q-function approximation and show great performances in complex tasks (see more details about related work in Section C). In practice, the radius d is not directly tunable: it depends on the underlying density and geometry of the dataset and is unknown to the algorithm. Therefore, we adopt a retrieval mechanism with size parameter k to practically controls locality.

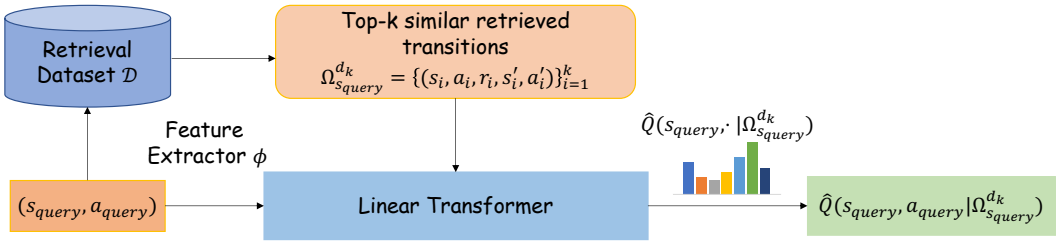


Figure 2: An overview of In-Context Compositional Q-Learning (ICQL). Given a query state-action pair $(s_{\text{query}}, a_{\text{query}})$, the model embeds it via our feature extractor ϕ, retrieves top-k similar transitions from a static offline dataset $\mathcal{D}$, and forms a local context set. A local linear Q-function approximation $\hat{Q}(s, a|\Omega^{d_k}_{s_{\text{query}}}) = w_s(\Omega^{d_k}_{s_{\text{query}}})^\top \phi(s, a)$ defined in Definition 3.1 is then fitted using the retrieved context $\Omega^{d_k}_{s_{\text{query}}}$ defined in Section 3.2, and used to update the actor. This enables compositional reasoning over local subtasks without requiring explicit subtask labels.

3.2 Retrieval Methods

In this section, we will introduce the approach to retrieve the transitions from the offline dataset $\mathcal{D}$. We mainly focus on state-similar retrieval, random retrieval and state-similar-with-high-reward retrieval. Each retrieval approach captures different coverage number of the local neighborhood $\Omega^{d_k}_{s_{\text{query}}}$ corresponding to the query state s_{query}. Both state-similar retrieval and state-similar-with-high-reward retrieval are supposed to capture more accurate and thorough local information from the local neighborhood Ω^d_s, and the main difference is that the state-similar retrieval is able to capture more diversity in the action space while the state-similar-with-high-rewards retrieval can ideally retrieve high-quality transitions. We will give the definition for state-similar retrieval in this section. Refer Section D to see more details and the definitions for the other two retrieval methods.

Definition 3.2 (State-Similar Retrieval)**.** Given the query state s_{query}, ICQL retrieves k many transitions based on the smallest l_2-distance between the retrieved state s_i and s_{query}, i.e.,

$$\overline{\Omega}^k_{s_{\text{query}}} \triangleq \left\{ (s_i, a_i, r_i, s'_i, a'_i) \in \mathcal{D} \middle| s_i \in \arg\text{top-k}\left\{ -\|s_{\text{query}} - s_i\|_2^2 \right\} \right\}. \tag{4}$$

Let us set $d_k^{s_{\text{query}}} \triangleq \max_{(s_i,a_i,r_i,s'_i,a'_i)\in\overline{\Omega}^k_{s_{\text{query}}}} \{\|s_{\text{query}} - s_i\|^2\}$, then we can conclude that $\overline{\Omega}^k_{s_{\text{query}}} = \Omega^{d_k^{s_{\text{query}}}}_{s_{\text{query}}}$. $d_k^{s_{\text{query}}}$ should be dependent on the query state s_{query}, but to make it easier for readers to follow, we will use d_k to represent $d_k^{s_{\text{query}}}$. Since our main ICQL utilizes the fixed state-similar retrieval method, we will use $\Omega^{d_k}_{s_{\text{query}}}$ to denote the retrieved context fed into the context of ICQL for notation consistency. In the next section, we will show how we use the transitions from $\Omega^{d_k}_{s_{\text{query}}}$ to learn the best local Q-function approximation $\hat{Q}_{\Omega^{d_k}_{s_{\text{query}}}}(s, a)$ for all $(s, a, r, s', a') \in \Omega^{d_k}_{s_{\text{query}}}$ through in-context learning.

3.3 In-context Compositional Q-Learning

Now, we are ready to show how we can learn compositional Q-functions through contextual inference. First, we will define the context-dependent weight function to estimate the optimal local weight vector w^*_s defined in Definition 3.1 corresponding to each state s.

Definition 3.3 (Context-dependent Weights)**.** The local weight function $w_s : \mathcal{P}(\Omega) \to \mathbb{R}^d$ is a context-dependent weight function inferred through in-context learning or retrieval-based adaptation, where $\mathcal{P}(\Omega) = \{A|A \subseteq \Omega\}$ is the power set of Ω and Ω contains all the possible transitions for some certain task.

We want to clarify that the offline dataset $\mathcal{D} \subseteq \Omega$. Based on Definition 3.3, there should exists some $\Omega^*_s \subseteq \Omega$ which leads to $w_s(\Omega^*_s) = w^*_s$. And it is not necessary that $\Omega^*_s \subseteq \mathcal{D}$. We can use different retrieval methods to cover Ω^*_s as much as possible to achieve a better weight approximation. Then for any query state s_{query} and action a_{query}, suppose $\Omega^{d_k}_{s_{\text{query}}}$ is the set collecting the k many retrieved

transitions by the state-similarity distance $d_{\min}$ from $\mathcal{D}$ defined in Section 3.2 and we feed $\Omega^d_{s_{\text{query}}}$ into the prompt matrix, we can learn a context-dependent Q-function approximation denoted as

$$\hat{Q}(s,a|\Omega^{d_k}_{s_{\text{query}}}) = w_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}})^T\phi(s,a) \tag{5}$$

to approximate $\hat{Q}_{\Omega^{d_k}_{s_{\text{query}}}}(s,a)$ defined in Equation (2). Next, we will explain how we can learn the local weight vector $w_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}})$ by in-context TD learning. The network updates $w(s_{\text{query}}|\Omega^{d_k}_{s_{\text{query}}})$ iteratively as for each retrieved transition $(s,a,r,s',a') \in \Omega^{d_k}_{s_{\text{query}}}$:

$$\begin{aligned}
&w^{\text{new}}_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}}) \\
=&w_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}}) + \alpha\Big(r + \gamma\hat{Q}(s',a'|\Omega^{d_k}_{s_{\text{query}}}) - \hat{Q}(s,a|\Omega^{d_k}_{s_{\text{query}}})\Big)\nabla_w\hat{Q}(s,a|\Omega^{d_k}_{s_{\text{query}}}) \\
=&w(s_{\text{query}}) + \alpha\Big(r + \gamma w_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}})^T\phi(s',a') - w_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}})^T\phi(s,a)\Big)\phi(s,a),
\end{aligned} \tag{6}$$

where α is the learning rate, the first equality is due to SARSA (Sutton & Barto, 2018) and the second equality is due to Equation (5). Please refer Section E to see more details about the construction of our linear transformers and the theorem to prove our proposed ICQL can implement in-context TD learning.

For training ICQL, we follow IQL (Kostrikov et al., 2021) to performs value iteration via expectile regression and policy extraction via advantaged-weighted regression. To be more specific, the critic loss is calculated with our local Q-function approximation:

$$\mathcal{L}_{\text{critic}} = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\rho_\tau\left(\hat{Q}(s,a|\Omega^{d_k}_s) - y\right)\right], \tag{7}$$

where $y = r + \gamma V(s'|\Omega^{d_k}_{s'}), V(s'|\Omega^{d_k}_{s'}) = \mathbb{E}_{a'\sim\pi}\left[\hat{Q}(s',a'|\Omega^{d_k}_{s'})\right]$, V is also a context dependent value estimator and $\rho_\tau(\cdot)$ denotes the expectile regression error. The policy is optimized via advantage-weighted regression, given the advantage based on local value estimation depending on current state and its retrieved similar states:

$$\mathcal{L}_{\text{policy}} = \mathbb{E}_{s\sim\mathcal{D}}\left[\mathbb{E}_{a\sim\pi}\left[\exp\left(\beta\cdot(\hat{Q}(s,a|\Omega^{d_k}_s) - V(s|\Omega^{d_k}_s))\right)\log\pi(a|s)\right]\right]. \tag{8}$$

After training, the extracted policy can be evaluated on its own without extra retrieval process or contextual inference.

3.4 Theoretical Analysis on ICQL

In this section, we analyze the theoretical properties of our algorithm ICQL. ICQL captures the compositional and local structures of complex decision-making tasks by enabling the Q-function to vary flexibly across different state regions. However, the performance of such local approximators depends critically on two factors:

(i) the expressiveness of the feature representation $\phi(s,a)$,

(ii) the accuracy of the learned weight function $w_s(\Omega^{d_k}_s)$ in approximating the optimal local weight w^*_s corresponding to the state s and the retrieved offline transition set $\Omega^{d_k}_s$.

To show that the performance of the greedy policy with respect to our ICQL is guaranteed to be near-optimal, we first need to derive point-wise and expected bounds on the local Q-function approximation error, highlighting how both approximation and weight estimation errors contribute to the total error. Building on these results, we further characterize how the approximation error propagates to policy sub-optimality through the performance difference lemma. These analyses provide theoretical justification for the importance of accurate local value estimation in achieving strong policy performance in offline RL settings. We will only show some necessary assumptions and the main theorem of near-optimal policy by ICQL in this section. Refer Section F to see more detailed and comprehensive proofs.

Assumption 3.1. Let $\phi : \mathcal{S}\times\mathcal{A} \to \mathbb{R}^d$ be a fixed feature map. We assume that for all $(s,a) \in \mathcal{S}\times\mathcal{A}$, the feature norm is bounded as $\|\phi(s,a)\| \le B_\phi$.

Algorithm 1 In-context Q-Learning (`ICQL`)

Input: Offline dataset $\mathcal{D}$, the number of retrieved transitions k, feature dimension d.
Initialize: Linear transformer TF_θ^Q with parameters θ, feature extractor ϕ.
Sample trajectory $\{(s_i, a_i, r_i)\}_{i=0}^{T-1} \sim \mathcal{D}$.
For each query state s_i, retrieve k sample states $s_i^0, \cdots, s_i^{k-1}$ based on state-similar retrieval method defined in Definition 3.2 and extract each of the corresponding transitions $\{(s_i^j, a_i^j, r_i^j, s_i'^j, a_i'^j)\}_{j=0}^{k-1}$.
//In-context Q value estimation.
for $t = 0, \ldots, T-1$ **do**
    Construct the input prompt matrix Z_t by Equation (24).
    $\hat{Q}_t \leftarrow TF_\theta^Q(Z_t)[2d+1, k+1]$ by Equation (16).
end for
Update the parameters θ, ϕ based on Equation (7) and Equation (8).

Remark 3.2. Assumption 3.1 is commonly used in previous research (Wang & Zou, 2020; Bhandari et al., 2018; Shen et al., 2020). In our experiments, we use tanh activation function at the last layer of our feature extractor ϕ, which means each component of the feature vector $\phi(s, a)$ is bounded by the positive constant 1. Hence, we can conclude that $\|\phi(s, a)\| \leq d$, where d is the feature dimension. This remark validates our Assumption 3.1.

Assumption 3.3 (Set Coverage)**.** For each query state $s_{\text{query}} \in \mathcal{S}$, let $\Omega^*_{s_{\text{query}}}$ denote the ideal local transition set defined in Section 3.3. Suppose the retrieved set $\Omega^{d_k}_{s_{\text{query}}}$ satisfies

$$\kappa_{s_{\text{query}}} \triangleq \frac{\left|\Omega^{d_k}_{s_{\text{query}}} \cap \Omega^*_{s_{\text{query}}}\right|}{|\Omega^*_{s_{\text{query}}}|} \geq \sigma, \tag{9}$$

for some coverage ratio $\sigma \in (0, 1]$. Equivalently, at least $m = \sigma|\Omega^*_{s_{\text{query}}}|$ transitions from $\Omega^*_{s_{\text{query}}}$ are contained in $\Omega^{d_k}_{s_{\text{query}}}$.

Remark 3.4. We use Assumption 3.3 to claim how many transitions from $\Omega^*_{s_{\text{query}}}$ can be covered by our retrieved set $\Omega^{d_k}_{s_{\text{query}}}$. This type of coverage condition is standard in nonparametric regression (Györfi et al., 2002; Devroye et al., 1996; Cover & Hart, 1967; Kpotufe, 2011) and has also been widely adopted in the analysis of offline RL through concentrability or coverage coefficients (Munos, 2003; 2007; Antos et al., 2008; Chen et al., 2019; Xie et al., 2021). The distance d_k and which retrieval method is used should affect the value κ_s. We show the ablation study on the number of transitions retrieved and the retrieval method in Section 4.3.

We now show our main theorem that the performance of the greedy policy with respect to the learned local Q-function approximation $\hat{Q}(s, a|\Omega^{d_k}_{s_{\text{query}}})$ is guaranteed to be near-optimal.

Theorem 3.5 (Policy Performance Gap)**.** *Suppose Assumptions 3.1 and 3.3 hold, and the learned policy π is greedy with respect to $\hat{Q}(s, a|\Omega^{d_k}_s)$. Then, with probability at least $1-\delta$, the performance gap is bounded as*

$$J(\pi^*) - J(\pi) \;\leq\; \frac{2}{1-\gamma}\, \mathbb{E}_{s\sim d^\pi}\left[\varepsilon^s_{approx}(1 + B_\phi) + CB_\phi\sqrt{\frac{d + \log(1/\delta)}{\sigma\,|\Omega^{d_k}_s|}}\right], \tag{10}$$

where $C > 0$ depends on B_ϕ, B_r and the conditioning of the local Gram matrix.

Proof. See more details in Section F.1. □

4 Experiments

4.1 Environments and Datasets

We evaluate our method on a diverse set of continuous control benchmarks from the D4RL suite (Fu et al., 2020), which includes three types of offline reinforcement learning environments:

Mujoco tasks (e.g., `HalfCheetah-Medium`) are standard locomotion environments based on MuJoCo (Todorov et al., 2012), featuring smooth dynamics and dense rewards. These tasks are commonly used to assess sample efficiency and stability.

Adroit tasks (e.g., `Pen-Human`) involve high-dimensional dexterous manipulation using a 24-DoF robotic hand. The action spaces are complex and the datasets are collected from human demonstration or behavior imitation, making them challenging due to limited action coverage.

Kitchen tasks (e.g., `Kitchen-Complete`) are long-horizon goal-conditioned tasks that require solving compositional subtasks (e.g., turning on lights, opening cabinets). These tasks emphasize multi-stage behavior and compositional reasoning.

4.2 Main Results

We compare our method against five widely adopted offline RL algorithms: `BC`, `DT` (Chen et al., 2021b), `TD3+BC` (Fujimoto & Gu, 2021), `CQL` (Kumar et al., 2020) and `IQL` (Kostrikov et al., 2022). These baselines represent two complementary paradigms: the first three represent policy constraints, and the last two represents value regularization. The experiment results are shown in Table 1.

Table 1: Performance comparison across Mujoco, Adroit, and Kitchen tasks. Average and standard deviation of scores are reported over 5 random seeds.

Mujoco Tasks	**BC**	**DT**	**TD3+BC**	**CQL**	**IQL**	**ICQL(Ours)**	**Gain(%)**
Walker2d-Medium-Expert-v2	107.5	70.7	109.2	98.7	109.8	**113.3**$_{\pm 2.0}$	3.1%
Walker2d-Medium-v2	75.3	70.2	77.0	79.2	71.5	**80.3**$_{\pm 5.2}$	1.4%
Walker2d-Medium-Replay-v2	26.0	54.8	41.5	77.2	61.0	**81.9**$_{\pm 5.4}$	6.1%
Hopper-Medium-Expert-v2	52.5	57.5	78.2	105.4	98.5	**108.8**$_{\pm 4.5}$	3.2%
Hopper-Medium-v2	52.9	57.1	53.5	58.0	**63.3**	62.6 $_{\pm 7.9}$	-1.5%
Hopper-Medium-Replay-v2	18.1	65.8	59.4	95.0	82.4	**96.4**$_{\pm 4.9}$	1.5%
HalfCheetah-Medium-Expert-v2	55.2	70.8	62.8	62.4	83.4	**89.1**$_{\pm 4.2}$	6.8%
HalfCheetah-Medium-v2	42.6	42.8	43.1	44.4	42.5	**45.9**$_{\pm 0.3}$	3.5%
HalfCheetah-Medium-Replay-v2	36.6	39.5	41.8	**45.5**	38.9	44.7 $_{\pm 0.1}$	-1.8%
Average	51.9	58.8	62.9	74.0	72.4	**80.3**	8.6%
Adroit Tasks	**BC**	**DT**	**TD3+BC**	**CQL**	**IQL**	**ICQL**	**Gain(%)**
Pen-Human-v1	63.9	79.5	64.6	37.5	**89.5**	85.6$_{\pm 5.6}$	-4.3%
Pen-Cloned-v1	37.0	74.0	76.8	39.2	4.9	**89.4**$_{\pm 4.8}$	5.4%
Hammer-Human-v1	1.2	1.7	1.5	4.4	**7.2**	3.7$_{\pm 3.2}$	-49.4%
Hammer-Cloned-v1	0.6	3.7	1.8	2.1	0.5	**4.5**$_{\pm 5.5}$	23.4%
Door-Human-v1	2.0	5.5	0.2	9.9	9.8	**17.1**$_{\pm 5.5}$	73.1%
Door-Cloned-v1	0.0	3.2	-0.1	0.1	7.6	**11.7**$_{\pm 4.4}$	53.6%
Average	17.45	27.9	24.2	15.5	33.2	**35.3**	6.3%
Kitchen Tasks	**BC**	**DT**	**TD3+BC**	**CQL**	**IQL**	**ICQL**	**Gain(%)**
Kitchen-Complete-v0	65.0	52.5	57.5	43.8	59.2	**79.3**$_{\pm 2.1}$	22.0%
Kitchen-Mixed-v0	51.5	**60.0**	53.5	51.0	53.3	59.5$_{\pm 6.0}$	-0.8%
Kitchen-Partial-v0	38.0	55.0	46.7	49.8	45.8	**61.5**$_{\pm 5.8}$	11.8%
Average	51.5	55.8	52.6	48.2	52.8	**66.8**	16.4%

Results demonstrate that, on Mujoco tasks, `ICQL` outperforms second best baseline `CQL` by 8.6% on average. On Adroit tasks, `ICQL` improves `IQL` by 6.3%. Notably, on Kitchen task, `ICQL` achieves a **16.4% improvement** over `DT` on Kitchen tasks, highlighting the importance of compositional value estimation in environments with complex, multi-stage structure. However on Hammer-Human dataset, `ICQL` is inferior to two baseline methods, which may relate to the dataset quality issue. In `Hammer-Human`, the size of the dataset is smaller and the distance between query states and retrieved similar states are larger than those of `Hammer-Cloned`, making it harder for in-context learning. Overall, these results validate the general applicability of `ICQL` across both value-learning and actor-critic paradigms.

For investigating whether `ICQL` can produce more accurate value estimation than baseline methods, we conduct analysis on the learned Q function by comparing the Q prediction among `ICQL`, `IQL` and online RL method `SAC`. We plot their Q estimations of the same set of offline dataset entries, and leverage t-SNE for showing their respective Q-estimate distribution over the same state space. Figure 3 shows the results on `Walker2d-Medium` dataset, where `ICQL` shares an approximately 69%

similarity with SAC on Q estimation, while IQL can only achieve a similarity score about 0.29. This indicates that the superior performance of ICQL on IQL comes from a better Q estimation, ensured by local Q function estimation, over the noisy dataset.

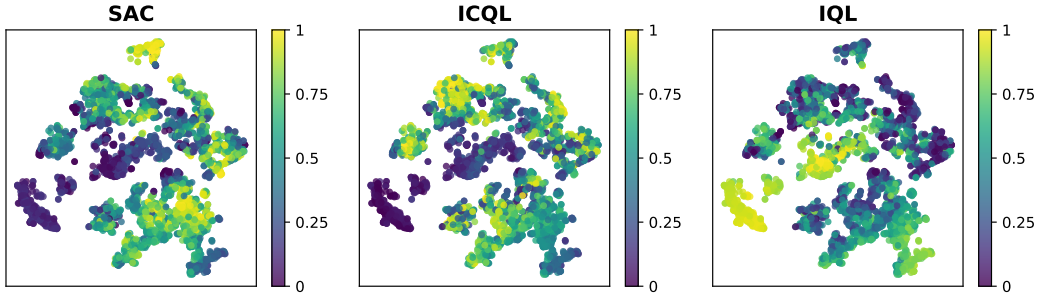
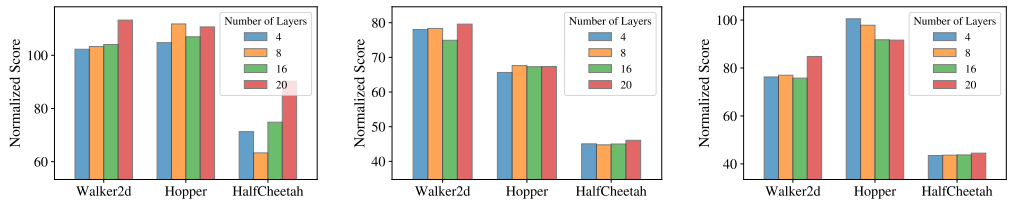


Figure 3: Q-value distribution on states after t-SNE dimension reduction, of Walker2d-Medium dataset. The partitioned value patterns support our hypothesis that Q-functions are inherently compositional, motivating localized value modeling.

4.3 Ablation Studies

4.3.1 Number of In-context Learning Layers

In this experiment, we investigate the effect of in-context learning steps, which is controlled by the number of layers in the in-context critic network. The number of layers are selected from $\{4, 8, 16, 20\}$. The experiments are conducted on Mujoco tasks and on the ICQL. Figure 4 displays the experiment outcomes and Table 7 provides further numerical results. From Figure 4, the normalized scores generally get higher as the number of layers get larger in most of the tasks, indicating that a larger number of layers may lead to more sufficient in-context value-learning. While the phenomenon is not obvious in Hopper tasks, one possible reason is the significant distribution shift in Hopper environment due to the high variance of transitions dynamics.

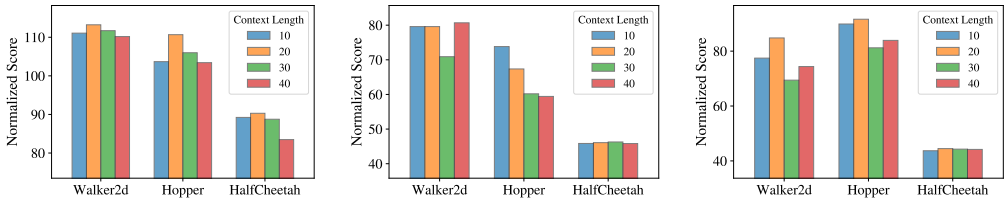


(a) Medium-Expert (b) Medium (c) Medium-Replay

Figure 4: Normalized scores of different number of in-context learning layers on Mujoco tasks. Each color represents different number of layers, and the y-axis represents the normalized score.

4.3.2 Influence of Context Length

In this experiment, we investigate the effect of context lengths in ICQL. The context lengths are selected from $\{10, 20, 30, 40\}$. As shown in Figure 5, a context length of 20 yields the generally best performance for in-context TD-learning in Gym tasks, where too long or too short context lengths lead to sub-optimal results. Thess results provide evidence that the "locality" of context is crucial for in-context learning performance. While the context lengths get longer, the distance between query state and context transitions also gets larger, which may break the "local" definition and bring noise into the in-context learning process. Detailed numerical results are shown in Table 7.



(a) `Medium-Expert` (b) `Medium` (c) `Medium-Replay`

Figure 5: Normalized scores of context lengths on Mujoco tasks. Each color represents different context lengths, and the y-axis represents the normalized score.

4.3.3 Context Retrieval Strategies

In this experiment, we investigate the impact of retrieval quality, by applying different context retrieval strategies on `ICQL`. Besides the standard **State-Similar Retrieval**, we compare two extra retrieval strategies: (1) **Random Retrieval**, which selects transitions uniformly at random from the offline dataset; and (2) **State-Similar-with-High-Rewards Retrieval**, which further filters the similar-state candidates by selecting those with higher rewards. The definitions of these three retrieval methods are defined in Sections 3.2 and D.

Our results show that the **Random Retrieval** performs poorly and leads to unstable training across environments, highlighting the importance of context relevance. The **State-Similar Retrieval** yields overall strong and consistent performance, demonstrating the benefit of local state-based context construction. Interestingly, in certain tasks with lower data quality, such as `walker2d-medium` and `door-human`, the **State-Similar-with-High-Rewards Retrieval** outperforms others. This suggests that incorporating reward information during retrieval can help identify more informative transitions, leading to better Q-function estimation in noisy or suboptimal datasets.

Table 2: Ablation study on retrieval strategies used in `ICQL`. We compare three variants: **Random Retrieval**, **State-Similar Retrieval**, and **State-Similar-with-High-Rewards Retrieval**.

Dataset	Random	State-Similar	State-Similar-with-High-Rewards
Walker2d-Medium-v2	78.14	79.59	**83.86**
Walker2d-Medium-Replay-v2	67.45	**84.81**	75.12
Hopper-Medium-v2	**74.14**	67.36	59.93
Hopper-Medium-Replay-v2	81.04	**91.63**	90.82
HalfCheetah-Medium-v2	45.53	46.08	**46.38**
HalfCheetah-Medium-Replay-v2	43.35	**44.48**	43.15
Pen-Human-v1	75.10	84.37	**84.82**
Hammer-Human-v1	1.42	2.05	**4.39**
Door-Human-v1	11.99	12.89	**15.59**
Kitchen-Complete-v0	70.00	**80.00**	71.25
Kitchen-Mixed-v0	53.75	**62.50**	60.00
Kitchen-Partial-v0	47.5	**62.50**	50.00

5 Conclusion and Future Work

We introduced `ICQL`, a novel offline RL framework that casts value estimation as an in-context inference problem using linear attention. By retrieving local transitions and fitting context-dependent local Q-functions, `ICQL` enables compositional reasoning without requiring subtask supervision. We provide theoretical guarantees to derive a near-optimal policy based on `ICQL` via greedy action extraction. Experiments show that `ICQL` achieves strong performance gains and provides closer value estimation to online reinforcement algorithms. These results highlight the potential of in-context learning as a powerful inductive bias for offline reinforcement learning. While the methodology of `ICQL` is agnostic to the distance metric, the quality of retrieval stands as a practical concern for

complex, high-dimensional state space. An important and promising direction for future work is incorporating ICQL with more sophisticated retrieval methods, such as pre-trained state encoders or value-aware learnable retriever.

REFERENCES

Marcin Andrychowicz, Dwight Crow, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 5048–5058, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/453fadbd8a1a3af50a9df4df899537b5-Abstract.html.

András Antos, Csaba Szepesvári, and Rémi Munos. Learning near-optimal policies with bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, 71:89–129, 2008.

Rushiv Arora. Hierarchical universal value function approximators, 2024. URL https://arxiv.org/abs/2410.08997.

Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In Satinder Singh and Shaul Markovitch (eds.), *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pp. 1726–1734. AAAI Press, 2017. doi: 10.1609/AAAI.V31I1.10916. URL https://doi.org/10.1609/aaai.v31i1.10916.

Glen Berseth, Daniel Geng, Coline Manon Devin, Nicholas Rhinehart, Chelsea Finn, Dinesh Jayaraman, and Sergey Levine. Smirl: Surprise minimizing reinforcement learning in unstable environments. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=cPZOyoDloxl.

Jalaj Bhandari, Daniel Russo, and Raghav Singal. A finite time analysis of temporal difference learning with linear function approximation. In *Conference on learning theory*, pp. 1691–1692. PMLR, 2018.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling. 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. 15084–15097, 2021a. URL https://proceedings.neurips.cc/paper/2021/hash/7f489f642a0ddb10272b5c31057f0663-Abstract.html.

Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, Pieter Abbeel, Aravind Srinivas, and Igor Mordatch. Decision transformer: Reinforcement learning via sequence modeling, 2021b.

Xi Chen, Nan Jiang, and Alekh Agarwal. Information-theoretic considerations in batch reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 1049–1058, 2019.

Thomas M Cover and Peter E Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(1):21–27, 1967.

Luc Devroye, László Györfi, and Gábor Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

Thomas G. Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *J. Artif. Intell. Res.*, 13:227–303, 2000. doi: 10.1613/JAIR.639. URL https://doi.org/10.1613/jair.639.

Simon Shaolei Du, Sham M. Kakade, Ruosong Wang, and Lin F. Yang. Is a good representation sufficient for sample efficient reinforcement learning? *ArXiv*, abs/1910.03016, 2019. URL https://api.semanticscholar.org/CorpusID:203902511.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=SJx63jRqFm.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement 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. 20132–20145, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/a8166da05c5a094f7dc03724b41886e5-Abstract.html.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2052–2062. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/fujimoto19a.html.

László Györfi, Michael Kohler, Adam Krzyżak, and Harro Walk. *A Distribution-Free Theory of Nonparametric Regression*. Springer, 2002.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=cp5PvcI6w8_.

Christian D. Hubbs, Hector D. Perez, Owais Sarwar, Nikolaos V. Sahinidis, Ignacio E. Grossmann, and John M. Wassick. Or-gym: A reinforcement learning library for operations research problems, 2020. URL https://arxiv.org/abs/2008.06319.

Natasha Jaques, Asma Ghandeharioun, Judy Hanwen Shen, Craig Ferguson, Àgata Lapedriza, Noah Jones, Shixiang Gu, and Rosalind W. Picard. Way off-policy batch deep reinforcement learning of implicit human preferences in dialog. *CoRR*, abs/1907.00456, 2019. URL http://arxiv.org/abs/1907.00456.

Sham M. Kakade, Michael Kearns, and John Langford. Exploration in metric state spaces. In *International Conference on Machine Learning*, 2003. URL https://api.semanticscholar.org/CorpusID:3713729.

Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey Levine. Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation. *ArXiv*, abs/1806.10293, 2018. URL https://api.semanticscholar.org/CorpusID:49470584.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning, 2021. URL https://arxiv.org/abs/2110.06169.

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

Samory Kpotufe. k-nn regression adapts to local intrinsic dimension. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 24, pp. 729–737, 2011.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/c2073ffa77b5357a498057413bb09d3a-Paper.pdf.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191, 2020.

Sascha Lange, Thomas Gabel, and Martin Riedmiller. *Batch Reinforcement Learning*, pp. 45–73. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-27645-3. doi: 10.1007/978-3-642-27645-3_2. URL https://doi.org/10.1007/978-3-642-27645-3_2.

Michael Laskin, Luyu Wang, Junhyuk Oh, Emilio Parisotto, Stephen Spencer, Richie Steigerwald, DJ Strouse, Steven Stenberg Hansen, Angelos Filos, Ethan Brooks, Maxime Gazeau, Himanshu Sahni, Satinder Singh, and Volodymyr Mnih. In-context reinforcement learning with algorithm distillation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=hy0a5MMPUv.

Jonathan Lee, Annie Xie, Aldo Pacchiano, Yash Chandak, Chelsea Finn, Ofir Nachum, and Emma Brunskill. Supervised pretraining can learn in-context reinforcement learning. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/8644b61a9bc87bf7844750a015feb600-Abstract-Conference.html.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *CoRR*, abs/2005.01643, 2020. URL https://arxiv.org/abs/2005.01643.

Qiyang Li, Zhiyuan Zhou, and Sergey Levine. Reinforcement learning with action chunking. *arXiv preprint arXiv:2507.07969*, 2025.

Yixiu Mao, Qi Wang, Yun Qu, Yuhang Jiang, and Xiangyang Ji. Doubly mild generalization for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 37:51436–51473, 2024.

Nina Mazyavkina, Sergey Sviridov, Sergei Ivanov, and Evgeny Burnaev. Reinforcement learning for combinatorial optimization: A survey. *Computers & Operations Research*, 134:105400, 2021. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2021.105400. URL https://www.sciencedirect.com/science/article/pii/S0305054821001660.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=vhFu1Acb0xb.

Subhojyoti Mukherjee, Josiah P. Hanna, Qiaomin Xie, and Robert D. Nowak. Pretraining decision transformers with reward prediction for in-context multi-task structured bandit learning. *CoRR*, abs/2406.05064, 2024. doi: 10.48550/ARXIV.2406.05064. URL https://doi.org/10.48550/arXiv.2406.05064.

Rémi Munos. Error bounds for approximate policy iteration. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*, pp. 560–567, 2003.

Rémi Munos. Performance bounds in l_p-norm for approximate value iteration. *SIAM Journal on Control and Optimization*, 46(2):541–561, 2007.

Ofir Nachum, Shixiang Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 3307–3317, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/e6384711491713d29bc63fc5eeb5ba4f-Abstract.html.

Seohong Park, Qiyang Li, and Sergey Levine. Flow q-learning. *arXiv preprint arXiv:2502.02538*, 2025.

Ronald E. Parr, Lihong Li, Gavin Taylor, Christopher Painter-Wakefield, and Michael L. Littman. An analysis of linear models, linear value-function approximation, and feature selection for reinforcement learning. In *International Conference on Machine Learning*, 2008. URL https://api.semanticscholar.org/CorpusID:11483966.

Pascal Poupart, Craig Boutilier, Relu Patrascu, and Dale Schuurmans. Piecewise linear value function approximation for factored mdps. In *AAAI/IAAI*, 2002. URL https://api.semanticscholar.org/CorpusID:8801238.

Scott E. Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, Tom Eccles, Jake Bruce, Ali Razavi, Ashley Edwards, Nicolas Heess, Yutian Chen, Raia Hadsell, Oriol Vinyals, Mahyar Bordbar, and Nando de Freitas. A generalist agent. *Trans. Mach. Learn. Res.*, 2022, 2022. URL https://openreview.net/forum?id=1ikK0kHjvj.

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

Thomas Schmied, Fabian Paischer, Vihang Patil, Markus Hofmarcher, Razvan Pascanu, and Sepp Hochreiter. Retrieval-augmented decision transformer: External memory for in-context rl. *arXiv preprint arXiv:2410.07071*, 2024.

Han Shen, Kaiqing Zhang, Mingyi Hong, and Tianyi Chen. Asynchronous advantage actor critic: Non-asymptotic analysis and linear speedup. 2020.

Lingfeng Shen, Aayush Mishra, and Daniel Khashabi. Position: Do pretrained transformers learn in-context by gradient descent? In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=WsawczEqO6.

Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT press, 2nd edition, 2018.

Denis Tarasov, Vladislav Kurenkov, Alexander Nikulin, and Sergey Kolesnikov. Revisiting the minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36:11592–11620, 2023.

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

Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 35151–35174. PMLR, 2023. URL https://proceedings.mlr.press/v202/von-oswald23a.html.

Jinyu Wang, Jingjing Fu, Rui Wang, Lei Song, and Jiang Bian. Pike-rag: specialized knowledge and rationale augmented generation, 2025a. URL https://arxiv.org/abs/2501.11551.

Jiuqi Wang, Ethan Blaser, Hadi Daneshmand, and Shangtong Zhang. Transformers can learn temporal difference methods for in-context reinforcement learning. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025b. URL https://openreview.net/forum?id=Pj06mxCXPl.

Tianshi Wang, Qikai Yang, Ruijie Wang, Dachun Sun, Jinyang Li, Yizhuo Chen, Yigong Hu, Chaoqi Yang, Tomoyoshi Kimura, Denizhan Kara, and Tarek F. Abdelzaher. Fine-grained control of generative data augmentation in iot sensing. In *Neural Information Processing Systems*, 2024. URL https://api.semanticscholar.org/CorpusID:276184922.

Xiangjun Wang, Junxiao Song, Penghui Qi, Peng Peng, Zhenkun Tang, Wei Zhang, Weimin Li, Xiongjun Pi, Jujie He, Chao Gao, Haitao Long, and Quan Yuan. Scc: an efficient deep reinforcement learning agent mastering the game of starcraft ii. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10905–10915. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/wang21v.html.

Yue Wang and Shaofeng Zou. Finite-sample analysis of greedy-gq with linear function approximation under markovian noise. In *Conference on Uncertainty in Artificial Intelligence*, pp. 11–20. PMLR, 2020.

Yifan Wu, George Tucker, and Ofir Nachum. Behavior regularized offline reinforcement learning. *CoRR*, abs/1911.11361, 2019. URL http://arxiv.org/abs/1911.11361.

Tengyang Xie, Yuzhe Ma, Zhuoran Yang, and Zhaoran Wang. Bellman-consistent pessimism for offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 6683–6694, 2021.

Dong Yin, Botao Hao, Yasin Abbasi-Yadkori, Nevena Lazić, and Csaba Szepesvári. Efficient local planning with linear function approximation. In Sanjoy Dasgupta and Nika Haghtalab (eds.), *Proceedings of The 33rd International Conference on Algorithmic Learning Theory*, volume 167 of *Proceedings of Machine Learning Research*, pp. 1165–1192. PMLR, 29 Mar–01 Apr 2022. URL https://proceedings.mlr.press/v167/yin22a.html.

Wenhao Zhao, Qiushui Xu, Linjie Xu, Lei Song, Jinyu Wang, Chunlai Zhou, and Jiang Bian. Unveiling markov heads in pretrained language models for offline reinforcement learning, 2025. URL https://arxiv.org/abs/2409.06985.

LLM USAGE STATEMENT

LLMs were used to aid the writing and polishing of the manuscript.

APPENDIX

A MORE EXPLANATIONS ABOUT COMPOSITIONAL Q-FUNCTIONS

We observed similar results when replacing return-to-go with reward or Q-values estimated by an online reinforcement learning-trained action-value function, which further strengthens our motivation. Taking Figure 1(c) as an example, which exhibits the most pronounced state clustering structure. We visualize randomly sampled states within neighboring regions. Dividing the space into four quadrants, we observe that: (a) States in the first quadrant are primarily associated with moving the kettle on the stove, (b)The second quadrant corresponds mainly to interacting with the light switch, (c) The third quadrant mostly involves manipulating the cabinet, and (d) the fourth quadrant includes states related to operating the microwave. These observations validate the motivation that similar states may share the same subtask to finish that it might be beneficial utilizing nearby context for Q-function estimation. Our experiments also show that ICQL has largely boosted performance on Kitchen tasks.

B PRELIMINARY

B.1 REINFORCEMENT LEARNING

We consider an infinite-horizon Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, p_0, p_{\text{MDP}}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote finite state and action spaces, respectively. The reward function is $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and the transition dynamics are governed by $p_{\text{MDP}}(s'|s, a)$, which denotes the probability of transitioning to state s' from state s after taking action a. The initial state distribution is $p_0 : \mathcal{S} \to [0, 1]$, and $\gamma \in [0, 1)$ is the discount factor.

At each timestep t, the agent observes state s_t, selects an action $a_t \sim \pi(\cdot|s_t)$ according to a stochastic policy $\pi : \mathcal{A} \times \mathcal{S} \to [0, 1]$, receives a reward $r_t = \mathcal{R}(s_t, a_t)$, and transitions to the next state $s_{t+1} \sim p_{\text{MDP}}(\cdot|s_t, a_t)$. This interaction generates trajectories of the form $(s_0, a_0, r_0, s_1, a_1, r_1, \dots)$.

Given a policy π, the associated Q-function and value function quantify the expected cumulative discounted rewards starting from state-action pair (s_t, a_t) and state s_t, respectively:

$$Q^\pi(s_t, a_t) \triangleq \mathbb{E}_{a_{t+1}, a_{t+2}, \dots \sim \pi} \left[\sum_{i=0}^{\infty} \gamma^i \mathcal{R}(s_{t+i+1}, a_{t+i+1}) | s_t, a_t \right], \tag{11a}$$

$$V^\pi(s_t) \triangleq \mathbb{E}_{a_t \sim \pi(\cdot|s_t)} \left[Q^\pi(s_t, a_t) \right]. \tag{11b}$$

The Q-function satisfies the *Bellman Expectation Equation*:

$$Q^\pi(s, a) = \mathcal{R}(s, a) + \gamma \, \mathbb{E}_{s' \sim p_{\text{MDP}}(\cdot|s,a)} \left[V^\pi(s') \right]. \tag{12}$$

Similarly, the value function satisfies:

$$V^\pi(s) = \mathbb{E}_{a \sim \pi(\cdot|s)} \left[Q^\pi(s, a) \right]. \tag{13}$$

The goal of reinforcement learning is to learn a policy $\pi_\theta(a|s)$ that maximizes the expected cumulative discounted rewards. The optimal value functions satisfy the *Bellman Optimality Equations*:

$$Q^*(s, a) = \mathcal{R}(s, a) + \gamma \, \mathbb{E}_{s' \sim p_{\text{MDP}}(\cdot|s,a)} \left[\max_{a'} Q^*(s', a') \right], \tag{14a}$$

$$V^*(s) = \max_{a \in \mathcal{A}} Q^*(s, a). \tag{14b}$$

In the offline setting, rather than interacting with the environment, the agent is provided with a fixed dataset $\mathcal{D} = \{(s, a, r, s')\}$, collected by a behavior policy π_β. Offline RL algorithms aim to learn an effective policy entirely from this static dataset $\mathcal{D}$, without any further environment interaction. A key challenge in offline RL is the *distributional shift* (Kumar et al., 2019; Jaques et al., 2019; Levine et al., 2020; Wu et al., 2019) between the learned policy π and the behavior policy π_β, which often leads to overestimation and poor generalization when estimating Q-values for out-of-distribution state-action pairs.

B.2 IN-CONTEXT LEARNING WITH LINEAR ATTENTIONS

Recently, there has been significant interest in understanding the theoretical capabilities of in-context learning with linear attention mechanisms (Wang et al., 2025b), particularly in the context of random instances of linear regression and simple classification tasks. We will formally introduce these problem settings in this section. Throughout this paper, all vectors are treated as column vectors. We denote the identity matrix in $\mathbb{R}^n$ by I_n, and the $m \times n$ all-zero matrix by $0_{m \times n}$. For any matrix Z, we use $Z^\top$ to denote its transpose, and use both $\langle x, y \rangle$ and $x^\top y$ interchangeably to denote the inner product.

We define a prompt matrix $Z \in \mathbb{R}^{(d+1) \times (n+1)}$ as follows:

$$Z \triangleq \begin{bmatrix} z^{(0)} & z^{(1)} & \dots & z^{(n-1)} & z^{(n)} \end{bmatrix} = \begin{bmatrix} x^{(0)} & x^{(1)} & \dots & x^{(n-1)} & x^{(n)} \\ y^{(0)} & y^{(1)} & \dots & y^{(n-1)} & 0 \end{bmatrix}, \tag{15}$$

where $\{x^{(i)}, y^{(i)}\}_{i=0}^{n-1}$ are context examples, $x^{(n)}$ is the query input with its corresponding response value $y^{(n)}$ masked as zero, and each $x^{(i)} \in \mathbb{R}^d$ and $y^{(i)} \in \mathbb{R}$ for all $i = 0, \cdots, n$. Following (von Oswald et al., 2023), we define linear self-attention over the same prompt as

$$\text{LinAttn}(Z; P, G) \triangleq PZM\left(Z^\top GZ\right), \tag{16}$$

where $P, G \in \mathbb{R}^{(d+1)\times(d+1)}$ are learnable parameter matrices, and $M \in \mathbb{R}^{(n+1)\times(n+1)}$ is a fixed mask matrix defined as

$$M \triangleq \begin{bmatrix} I_n & 0_{n\times 1} \\ 0_{1\times n} & 0 \end{bmatrix}. \tag{17}$$

The goal of training linear transformers in this setting is to recover the unknown response variable corresponding to $x^{(n)}$, which is represented as zero in the prompt matrix Z. By appropriately constructing the parameter matrices P and G, the linear attention model in Equation (16) can successfully perform in-context learning for linear regression and simple classification tasks. However, the ability of such models to perform in-context learning for offline reinforcement learning remains poorly understood. And these analyses are purely theoretical and have not been empirically validated on practical tasks. Transformers can perform in-context supervised learning by mimicking gradient descent updates (von Oswald et al., 2023), and in-context reinforcement learning through TD-like methods via appropriately constructed linear attention mechanisms (Wang et al., 2025b). However, (Wang et al., 2025b) considers only the simplified setting of Markov Reward Processes (MRPs), where transitions and rewards depend solely on the current state, i.e., $s_{t+1} \sim p(\cdot|s_t)$ and $r_{t+1} = r(s_t)$, with time-dependent context representations. More precisely, their formulation assumes that each trajectory consists solely of temporally continuous steps. These restrictive assumptions do not hold in real-world decision-making problems, and their empirical results are limited to synthetic MRPs, which is hard to predict its performance on real-life RL tasks. To bridge this gap, we extend the analysis from MRPs to the more general MDP setting by estimating the state-action value function $Q(s, a)$ directly and removing the time dependency from the context representations.

C Other Related Work

Goal-conditioned and Hierarchical RL. Goal-conditioned methods such as `UVFA` (Schaul et al., 2015) and `HER` (Andrychowicz et al., 2017) condition policies or value functions on explicit goal inputs to facilitate generalization across tasks. Extensions to compositional settings further decompose Q-functions into subgoal components (Arora, 2024). However, these approaches assume access to goal specifications or subtask labels, which are typically unavailable in offline settings. `ICQL` addresses this limitation by learning Q-functions conditioned on retrieved transition contexts, eliminating the need for task supervision and enhancing sample efficiency. Hierarchical reinforcement learning decomposes tasks into subgoals or options, enabling temporal abstraction and subpolicy reuse. Classical methods such as `MAXQ` (Dietterich, 2000), `Option-Critic` (Bacon et al., 2017), and `HIRO` (Nachum et al., 2018) explicitly model subtask boundaries and learn separate value functions for each. While effective when task structure is known or discoverable, these methods often rely on subgoal specification or auxiliary termination conditions. In contrast, `ICQL` operates without predefined subtask structure and efficiently leverages offline data to rapidly converge to a provable accurate local value function approximation. Unsupervised RL methods such as `DIAYN` (Eysenbach et al., 2019) and `SMiRL` (Berseth et al., 2021) aim to discover diverse behaviors or latent subpolicies without external rewards or supervision. Although these methods can implicitly uncover structure, they are typically designed for unsupervised exploration or pretraining rather than for accurate value estimation in offline settings. `ICQL` instead focuses on precise local Q-function inference conditioned on retrieved experiences, thereby improving compositional generalization and training stability in the offline RL regime.

Linear Q-function Approximation. Linear Q-function approximation has been widely used in previous research (Yin et al., 2022; Du et al., 2019; Poupart et al., 2002; Parr et al., 2008). Metric MDPs (Kakade et al., 2003), which gives the definition of the Q-function according to the state distance metric, are a natural complement to more direct parametric assumptions on value functions and dynamics (Kakade et al., 2003). But none of them considers the local linear Q-function approximation based on the state distance metric. In our work, we focus on learning the better approximations of local value functions, while Kakade et al. (2003) formed an accurate approximation of the local

environment. We assume that for each local domain Ω_s^d, the local Q-function should have its own state-dependent local structure. This has been examined both theoretically and practically to give a better Q-function approximation and show great performances in complex tasks.

D Detailed Definitions of Retrieval Methods

Retrieval methods show great performance among a lot of domains (Wang et al., 2024; 2025a). In this section, we will show the definitions for the other two retrieval methods – random retrieval and state-similar-with-high-rewards retrieval.

Definition D.1 (Random Retrieval)**.** Given the query state s_{query}, randomly retrieved context for `ICQL` is defined as

$$\overline{\Omega}^{\text{random}}_{s_{\text{query}}} \triangleq \left\{(s_i, a_i, r_i, s_i', a_i') \in \mathcal{D} \middle| (s_i, a_i, r_i, s_i', a_i') \sim \mathcal{D} \right\}_{i=0}^{k-1}. \tag{18}$$

Definition D.2 (State-Similar-with-High-Rewards Retrieval)**.** Given the query state s_{query}, $\overline{\Omega}^{\text{high}}_{s_{\text{query}}}$ for `ICQL` is defined as k many transitions with the smallest l_2-distance between the retrieved state s_i and s_{query} and the highest transition reward r_i, i.e.,

$$\overline{\Omega}^{\text{high}}_{s_{\text{query}}} \triangleq \left\{(s_i, a_i, r_i, s_i', a_i') \in \overline{\Omega}^{k_s}_{s_{\text{query}}} \middle| (s_i, a_i, r_i, s_i', a_i') \in \arg\text{top-k}\,\{r_i\} \right\}, \tag{19}$$

where $\overline{\Omega}^{k_s}_{s_{\text{query}}}$ is defined in Equation (4).

For the retrieval methods defined in Definitions 3.2, D.1, and D.2, we can relate them to Equation (1) by simply letting $d_1 \triangleq \min_{(s_i,a_i,r_i,s_i',a_i') \in \overline{\Omega}^{k}_{s_{\text{query}}}} \left\{\|s_i - s_{\text{query}}\|_2\right\}$ and $d_2 \triangleq \min_{(s_i,a_i,r_i,s_i',a_i') \in \overline{\Omega}^{\text{top}}_{s_{\text{query}}}} \left\{\|s_i - s_{\text{query}}\|_2\right\}$. Therefore, we can conclude that $\overline{\Omega}^{k}_{s_{\text{query}}} \subseteq \Omega^{d_1}_{s_{\text{query}}}$ and $\overline{\Omega}^{\text{high}}_{s_{\text{query}}} \subseteq \Omega^{d_2}_{s_{\text{query}}}$, which implies that both state-similar retrieval and state-similar-with-high-reward retrieval can be bounded by some local neighborhood corresponding to the query state s_{query}.

E Designs of Linear Transformers for both Sparse-reward and Dense-reward RL Tasks

In this section, we will explain how our `ICQL` is constructed and how it can be extended to sparse-reward tasks. Due to the initialization $w_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}}) = 0$ for all s_{query} and Equation (6), we will observe that after one iteration update of the weight,

$$\begin{aligned} & w^{\text{new}}_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}}) \\ =& w_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}}) + \alpha\Big(r + \gamma w_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}})^T\phi(s',a') - w_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}})^T\phi(s,a)\Big)\phi(s,a) \\ =& \alpha r\phi(s,a) \end{aligned} \tag{20}$$

It leads to $w^{\text{new}}_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}}) \equiv 0$ when the tasks have sparse rewards, i.e., all the transition rewards r are equal to zero. It will lead to no weight update for `ICQL`. Hence, we propose a novel adapative

SARSA update rule for all the tasks augmented by Returns-to-go (RTGs), which is defined as

$$
\begin{aligned}
w^{\text{new}}_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}}) &= w_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}}) \\
&+ \alpha\Big[r + \gamma\Big(\frac{w_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}})^T\phi(s',a')}{\Big(w_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}})^T\phi(s',a') + \text{RTG}_{s'}\Big)} \cdot w_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}})^T\phi(s',a') \\
&+ \frac{\text{RTG}_{s'}}{\Big(w_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}})^T\phi(s',a') + \text{RTG}_{s'}\Big)} \cdot \text{RTG}_{s'}\Big) \\
&- \Big(\frac{w_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}})^T\phi(s,a)}{\Big(w(s_{\text{query}})^T\phi(s,a) + \text{RTG}_s\Big)} \cdot w_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}})^T\phi(s,a) \\
&+ \frac{\text{RTG}_s}{\Big(w_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}})^T\phi(s,a) + \text{RTG}_s\Big)} \cdot \text{RTG}_s\Big)\Big]\phi(s,a) \\
\approx & w_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}}) + \alpha\Big[r + \gamma\Big(\beta \cdot w_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}})^T\phi(s',a') + (1-\beta)\cdot \text{RTG}_{s'}\Big) \\
&- \Big(\beta \cdot w_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}})^T\phi(s,a) + (1-\beta)\cdot \text{RTG}_s\Big)\Big]\phi(s,a) \\
= & w_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}}) + \alpha\Big[\Big(r + \gamma(1-\beta)\cdot \text{RTG}_{s'} - (1-\beta)\text{RTG}_s\Big) \\
&+ \gamma\beta \cdot w_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}})^T\phi(s',a') - \beta \cdot w_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}})^T\phi(s,a)\Big]\phi(s,a),
\end{aligned}
\tag{21}
$$

where $\beta \in [0,1]$ is a task-dependent hyperparameter. We use the convex combination between $\hat{Q}(s',a'|\Omega^{d_k}_{s_{\text{query}}})$ and $\text{RTG}_{s'}$ to estimate each $Q_{\Omega^{d_k}_{s_{\text{query}}}}(s',a')$. To satisfy the construction in Equation (21), we will show our new design of input matrix, weight matrices for our `ICQL`. Given any query state s_{query} and N total many retrieved transitions in $\overline{\Omega}^{\text{random}}_{s_{\text{query}}}$. Using as shorthand $\phi_i \triangleq \phi(s_i,a_i)$ and $\phi'_i \triangleq \phi(s'_i,a'_i)$, the new input prompt matrix is define as

$$
Z_0 = \begin{bmatrix} \phi_0 & \cdots & \phi_{N-1} & \phi_{\text{query}} \\ \gamma\beta\phi'_0 & \cdots & \gamma\beta\phi'_{N-1} & 0 \\ r'_0 & \cdots & r'_{N-1} & 0 \end{bmatrix}, \tag{22}
$$

where $r'_i \triangleq r_i + \gamma(1-\beta)\cdot \text{RTG}_{s'_i} - (1-\beta)\text{RTG}_{s_i}$ for all $i = 0,\cdots,N-1$, and $\phi_{\text{query}} \triangleq \phi(s_{\text{query}}, a_{\text{query}})$ for any $a_{\text{query}} \in \mathcal{A}$. And for $\ell = 0,1,\cdots,L-1$, each linear transformer layer ℓ has weight matrices P_ℓ and G_ℓ defined as

$$
P_\ell \triangleq \begin{bmatrix} 0_{2d\times 2d} & 0_{2d\times 1} \\ 0_{1\times 2d} & 1 \end{bmatrix}, G_\ell \triangleq \begin{bmatrix} -C_\ell^T & C_\ell^T & 0_{d\times 1} \\ 0_{d\times d} & 0_{d\times d} & 0_{d\times 1} \\ 0_{1\times d} & 0_{1\times d} & 0 \end{bmatrix}, \tag{23}
$$

where all the matrices $\{C_\ell\}_{\ell=0}^{L-1}$ are trainable parameters.

Remark E.1. For Equation (22), when we set $\beta = 1$, Z_0 will recover the input prompt matrix for dense-reward tasks, which is defined as

$$
Z_0 = \begin{bmatrix} \phi_0 & \cdots & \phi_{N-1} & \phi_{\text{query}} \\ \gamma\phi'_0 & \cdots & \gamma\phi'_{N-1} & 0 \\ r_0 & \cdots & r_{N-1} & 0 \end{bmatrix} \tag{24}
$$

and the weight matrices P_ℓ and G_ℓ keep the same.

Next, we will prove how we can the weight update defined in Equation (6) by our design. First, we introduce the following lemma, which is motivated by the work of (Wang et al., 2025b) on MRPs.

Lemma E.2. *Consider the input Z_0 and matrix weights P_0 and Q_0, where*

$$
Z_0 = \begin{bmatrix} v_0^{(0)} & \cdots & v_0^{(N-1)} & v_0^{(N)} \\ \xi_0^{(0)} & \cdots & \xi_0^{(N-1)} & \xi_0^{(N)} \\ y_0^{(0)} & \cdots & y_0^{(N-1)} & y_0^{(N)} \end{bmatrix}, P_0 \doteq \begin{bmatrix} 0_{2d\times 2d} & 0_{2d\times 1} \\ 0_{1\times 2d} & 1 \end{bmatrix}, G_0 \doteq \begin{bmatrix} -C_0^T & C_0^T & 0_{d\times 1} \\ 0_{d\times d} & 0_{d\times d} & 0_{d\times 1} \\ 0_{1\times d} & 0_{1\times d} & 0 \end{bmatrix}, \tag{25}
$$

and $v^{(i)}, \xi^{(i)} \in \mathbb{R}^d$, $y^{(i)} \in \mathbb{R}$. *According to* $Z_1 \triangleq \mathrm{LinAttn}(Z_0; P_0, G_0) = P_0 Z_0 M (Z_0^T G_0 Z_0)$ *and let* $y_1^{(N)}$ *be the bottom right element of the next layer's output, i.e.,* $y_1^{(N)} \triangleq Z_1[2d+1, N+1]$, *it holds that* $y_1^{(N)} = -\langle \phi_N, w_1 \rangle$, *where*

$$w_1 = w_0 + \frac{1}{N} C_0 \sum_{j=0}^{N-1} (y_0^{(i)} + w_0^T \xi_0^{(i)} - w_0^T v_0^{(i)}) v_0^{(i)}. \tag{26}$$

Using the above lemma, we are ready to prove Theorem E.3.

Theorem E.3. *Consider the* L*-layer linear transformer following Equation* (16) *and all matrices* $\{P_\ell, G_\ell\}_{\ell=0}^L$, *mask matrix* M, *the input prompt matrix* Z_0 *are defined in Equations* (17), (23), *and* (24), *respectively. Then* $Z_\ell[2d+1, n+1]$, *the bottom right element of the* ℓ*-th layer's output, holds that* $Z_\ell[2d+1, n+1] = -\langle \phi_{query}, w^\ell_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}}) \rangle$, *where* $\{w^\ell_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}})\}$ *is defined as* $w^0_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}}) = 0$ *and for* $\ell \geq 0$

$$\begin{aligned} & w^{\ell+1}_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}}) \\ =& w^\ell_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}}) + \frac{1}{N} C_\ell \sum_{j=0}^{N-1} (r_j + \gamma w^\ell_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}})^T \phi'_j - w^\ell_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}})^T \phi_j) \phi_j. \end{aligned} \tag{27}$$

Proof. Let $v_0^{(i)} = \phi_i = \phi(s_i, a_i)$, $\xi_0^{(i)} = \gamma \phi'_i = \phi(s'_i, a'_i)$, $y_0^{(i)} = r_i$ for $i \in \{0, \cdots, N-1\}$ and $v_0^{(N)} = \phi_{\text{query}} = \phi(s_{\text{query}}, a_{\text{query}})$, $\xi_0^{(N)} = 0_{d \times 1}$, $y_0^{(N)} = 0$, we get

$$w^1_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}}) = w^0_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}}) + \frac{1}{N} C_0 \sum_{j=0}^{N-1} (r_i + \gamma w^0_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}})^T \phi'_i - w^0_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}})^T \phi_i) \phi_i,$$

which is the update rule for pre-conditioned SARSA. We also have

$$y_1^{(N)} = -\langle w^1_{s_{\text{query}}}(\Omega^{d_k}_{s_{\text{query}}}), \phi_{\text{query}} \rangle.$$

By induction on the number of layer ℓ, it completes our proof. □

F PROOFS

In this section, we first derive pointwise and expected bounds on the Q-function approximation error, highlighting how both approximation and weight estimation errors contribute to the total error. Building on these results, we further characterize how the approximation error propagates to policy suboptimality through the performance difference lemma. These analyses provide theoretical justification for the importance of accurate local value estimation in achieving strong policy performance, particularly in offline RL settings.

Theorem F.1 (Weight Error under Coverage)**.** *Suppose Assumption 3.3 holds, and that the feature vectors are bounded as* $\|\phi(s,a)\| \leq B_\phi$ *and rewards as* $|r| \leq B_r$. *Let* w_s^* *be the optimal local weight vector defined in Definition 3.1, and let* $w_s(\Omega_s^{d_k})$ *be the weight estimated from the retrieved set. Then with probability at least* $1-\delta$, *the following holds:*

$$\left\| w_s(\Omega_s^{d_k}) - w_s^* \right\| \;\leq\; C \left(\sqrt{\frac{d + \log(1/\delta)}{\sigma \, |\Omega_s^{d_k}|}} \;+\; \varepsilon^s_{\text{approx}} \right), \tag{28}$$

where $C > 0$ *is a constant depending on* B_ϕ, B_r *and the conditioning of the local Gram matrix, and* $\varepsilon^s_{\text{approx}}$ *is the local approximation error defined in Definition 3.1.*

Proof. Fix a query state s and its ideal local transition set Ω_s^*. By Definition 3.1, there exists a weight vector w_s^* such that

$$Q_{\Omega_s^{d_k}}(s,a) = {w_s^*}^\top \phi(s,a) + \varepsilon_s(s,a), \qquad |\varepsilon_s(s,a)| \leq \varepsilon^s_{\text{approx}} \tag{29}$$

for all $(s, a, r, s', a') \in \Omega_s^{d_k}$. By Assumption 3.3, the retrieved set $\Omega_s^{d_k}$ overlaps with the ideal set on at least $m = \sigma|\Omega_s^{d_k}|$ transitions. Denote this intersection as $\mathcal{D}_s^\sigma = \Omega_s^{d_k} \cap \Omega_s^*$. Thus the estimation of w_s^* from $\Omega_s^{d_k}$ is guaranteed to include at least m valid local transitions. Let $X \in \mathbb{R}^{m\times d}$ be the feature matrix of $\mathcal{D}_s^\alpha$, with columns $\phi(\bar{s}, \bar{a})$, and $y \in \mathbb{R}^m$ be the corresponding targets. Then

$$y = {w_s^*}^\top X + \xi, \tag{30}$$

where ξ collects the local approximation error, with $\|\xi\|_\infty \le \varepsilon^s_{\text{approx}}$. The estimator from the retrieved set is

$$w_s(\Omega_s^{d_k}) = \arg\min_w \frac{1}{|\Omega_s^{d_k}|} \sum_{(s_i,a_i)\in\Omega_s^{d_k}} \big(y_i - w^\top \phi(s_i, a_i)\big)^2. \tag{31}$$

Define the population moments on Ω_s^* as

$$G = \mathbb{E}_{\Omega_s^*}[\phi^\top \phi], \quad b = \mathbb{E}_{\Omega_s^*}[\phi^\top y]. \tag{32}$$

Let $\hat{G}, \hat{b}$ be the corresponding empirical moments on $\Omega_s^{d_k}$. Since at least $m = \sigma|\Omega_s^*|$ samples in $\Omega_s^{d_k}$ come from the true local set, standard matrix concentration implies that with probability at least $1-\delta$,

$$\|\hat{G} - G\| \le c_1 B_\phi^2 \sqrt{\tfrac{d+\log(1/\delta)}{\sigma|\Omega_s^{d_k}|}}, \tag{33}$$

$$\|\hat{b} - b\| \le c_2 B_\phi B_r \sqrt{\tfrac{d+\log(1/\delta)}{\sigma|\Omega_s^{d_k}|}}, \tag{34}$$

for universal constants $c_1, c_2 > 0$. The optimal weight satisfies ${w_s^*}^\top G = b$. The empirical solution satisfies $w_s(\Omega_s^{d_k})^\top \hat{G} = \hat{b}$ (up to residuals). Subtracting these systems gives

$$\|w_s(\Omega_s^{d_k}) - w_s^*\| \le \|G^{-1}\| \cdot \big(\|\hat{b} - b\| + \|\hat{G} - G\|\|w_s^*\|\big) + \varepsilon^s_{\text{approx}}. \tag{35}$$

Since G is well-conditioned, $\|G^{-1}\| \le 1/\mu$ for some $\mu > 0$. Substituting the concentration results yields

$$\|w_s(\Omega_s^{d_k}) - w_s^*\| \le C\sqrt{\tfrac{d+\log(1/\delta)}{\sigma|\Omega_s^{d_k}|}} + \varepsilon^s_{\text{approx}}, \tag{36}$$

where $C > 0$ depends on $B_\phi, B_r, \|w_s^*\|$ and μ. This is exactly the desired bound equation 28. □

Theorem F.2 (Pointwise Q-function Error)**.** *Suppose Assumption 3.1 and Assumption 3.3 hold. For any fixed $s \in \mathcal{S}$, with probability at least $1-\delta$, the pointwise error of the estimated Q-function satisfies*

$$\left|\hat{Q}(s, a|\Omega_s^{d_k}) - Q_{\Omega_s^{d_k}}(s, a)\right| \le \varepsilon^s_{\text{approx}}(1 + B_\phi) + CB_\phi\sqrt{\frac{d+\log(1/\delta)}{\sigma\,|\Omega_s^{d_{\min}}|}} \quad \forall (s, a, r, s', a') \in \Omega_s^{d_k}, \tag{37}$$

where $C > 0$ depends on B_ϕ, B_r and the conditioning of the local Gram matrix.

Proof. Fix $s \in \mathcal{S}$ and $a \in \mathcal{A}$. By definition,

$$\hat{Q}(s, a|\Omega_s^{d_k}) = w_s(\Omega_s^{d_k})^\top \phi(s, a), \qquad Q_{\Omega_s^{d_k}}(s, a) = {w_s^*}^\top \phi(s, a) + \varepsilon^s_{\text{approx}}. \tag{38}$$

Thus,

$$\left|\hat{Q}(s, a|\Omega_s^{d_k}) - Q_{\Omega_s^{d_k}}(s, a)\right| = \left|w_s(\Omega_s^{d_k})^\top \phi(s, a) - {w_s^*}^\top \phi(s, a) - \varepsilon^s_{\text{approx}}\right| \tag{39}$$

$$\le \|w_s(\Omega_s^{d_k}) - w_s^*\| \cdot \|\phi(s, a)\| + \varepsilon^s_{\text{approx}} \tag{40}$$

$$\le B_\phi \cdot \|w_s(\Omega_s^{d_k}) - w_s^*\| + \varepsilon^s_{\text{approx}}. \tag{41}$$

By Theorem F.1, with probability at least $1-\delta$,

$$\|w_s(\Omega_s^{d_k}) - w_s^*\| \le C\sqrt{\tfrac{d+\log(1/\delta)}{\sigma|\Omega_s^{d_{\min}}|}} + \varepsilon^s_{\text{approx}}. \tag{42}$$

Substituting this into the inequality above yields

$$\left|\hat{Q}(s, a|\Omega_s^{d_k}) - Q_{\Omega_s^{d_k}}(s, a)\right| \le CB_\phi\sqrt{\tfrac{d+\log(1/\delta)}{\sigma|\Omega_s^{d_k}|}} + \varepsilon^s_{\text{approx}}(1 + B_\phi), \tag{43}$$

which holds for all $(s, a, r, s', a') \in \Omega_s^{d_k}$. This proves equation 37. □

Corollary F.3 (Expected Q-function Error)**.** *Suppose Assumptions 3.1 and 3.3 hold. Let μ be a reference distribution over $(s,a) \in \mathcal{S} \times \mathcal{A}$, and let $\mu_{\mathcal{S}}$ be its marginal over states. Then, with probability at least $1-\delta$, the expected Q-function approximation error restricted to the retrieved set satisfies*

$$\begin{aligned} &\mathbb{E}_{(s,a)\sim\mu}\Big[\big|\hat{Q}(s,a|\Omega_s^{d_k}) - Q_{\Omega_s^{d_k}}(s,a)\big| \,\big|\, (s,a) \in \Omega_s^{d_k}\Big] \\ &\leq \mathbb{E}_{s\sim\mu_{\mathcal{S}}}\left[\varepsilon^s_{approx}(1+B_\phi) + CB_\phi\sqrt{\frac{d+\log(1/\delta)}{\sigma\,|\Omega_s^{d_k}|}}\right]. \end{aligned} \tag{44}$$

Proof. From Theorem F.2, for any $(s,a,r,s',a') \in \Omega_s^{d_k}$, we have

$$\big|\hat{Q}(s,a|\Omega_s^{d_k}) - Q_{\Omega_s^{d_k}}(s,a)\big| \;\leq\; \varepsilon^s_{\text{approx}}(1+B_\phi) + CB_\phi\sqrt{\tfrac{d+\log(1/\delta)}{\sigma|\Omega_s^{d_k}|}}. \tag{45}$$

Taking expectation over $(s,a)\sim\mu$, but restricted to $(s,a)\in\Omega_s^{d_k}$, and noting that the right-hand side depends only on s, we obtain

$$\begin{aligned} &\mathbb{E}_{(s,a)\sim\mu}\Big[\big|\hat{Q}(s,a|\Omega_s^{d_k}) - Q_{\Omega_s^{d_k}}(s,a)\big| \,\big|\, (s,a) \in \Omega_s^{d_k}\Big] \\ &\leq \mathbb{E}_{s\sim\mu_{\mathcal{S}}}\left[\varepsilon^s_{\text{approx}}(1+B_\phi) + CB_\phi\sqrt{\tfrac{d+\log(1/\delta)}{\sigma|\Omega_s^{d_k}|}}\right]. \end{aligned} \tag{46}$$

This proves the result. □

F.1 Proof of Theorem 3.5

Lemma F.4 (Performance Difference Lemma)**.** *Let π be a policy, and let d^π denote its discounted state distribution. Then the performance gap between π and the optimal policy π^* satisfies*

$$J(\pi^*) - J(\pi) = \frac{1}{1-\gamma}\,\mathbb{E}_{s\sim d^\pi, a\sim\pi}\Big[Q^*(s,a^*) - Q^*(s,a)\Big], \tag{47}$$

where $a^* = \arg\max_a Q^*(s,a)$.

Proof. From Equation (47), for any $s\in\mathcal{S}$,

$$\begin{aligned} Q^*(s,\pi^*(s)) - Q^*(s,\pi(s)) = \big(Q^*(s,\pi^*(s)) - \hat{Q}(s,\pi^*(s))\big) + \big(\hat{Q}(s,\pi^*(s)) - \hat{Q}(s,\pi(s))\big) \\ + \big(\hat{Q}(s,\pi(s)) - Q^*(s,\pi(s))\big). \end{aligned} \tag{48}$$

Since π is greedy w.r.t. $\hat{Q}$, the middle term is non-positive. Thus,

$$\begin{aligned} Q^*(s,\pi^*(s)) - Q^*(s,\pi(s)) &\leq |Q^*(s,\pi^*(s)) - \hat{Q}(s,\pi^*(s))| + |Q^*(s,\pi(s)) - \hat{Q}(s,\pi(s))| \\ &\leq 2\delta(s), \end{aligned} \tag{49}$$

where by Theorem F.2,

$$\delta(s) = \varepsilon^s_{\text{approx}}(1+B_\phi) + CB_\phi\sqrt{\tfrac{d+\log(1/\delta)}{\sigma|\Omega_s^{d_k}|}}. \tag{50}$$

Taking expectations in Equation (47) and applying Equation (49) yields

$$J(\pi^*) - J(\pi) \leq \frac{2}{1-\gamma}\,\mathbb{E}_{s\sim d^\pi}\big[\delta(s)\big], \tag{51}$$

which gives the desired bound equation 10. □

G ICQL VARIANTS FOR TD3+BC

In this section, we illustrate how to extend our method to TD3+BC (Fujimoto & Gu, 2021). TD3+BC introduces a simple behavior cloning regularization over value-based learning. This algorithms is easy to integrate with our framework, stable across diverse tasks, and serve as strong baselines in the literature. Their simplicity and effectiveness make them ideal testbeds for evaluating the impact of localized Q-function estimation, and together they offer sufficient coverage of common design choices in offline RL. Other algorithms can be similarly extended, but are omitted here for clarity and focus.

Our proposed ICQL can be seamlessly integrated into existing offline RL algorithms by replacing the global Q-function with a local, context-dependent estimator defined in Definition 3.1. We demonstrate this idea by instantiating ICQL with TD3+BC (see more details in our Algorithm 1).

ICQL-TD3+BC. TD3+BC uses a standard Bellman backup for the critic and augments the actor with behavior cloning. We again use the locally estimated $\hat{Q}(s,a)$ in both components. The critic loss is:

$$\mathcal{L}_{\text{critic}}^{\texttt{TD3+BC}} = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(\hat{Q}(s,a|\Omega_s^{d_k}) - y\right)^2\right], \tag{52}$$

where $y = r + \gamma \min_{i=1,2} \hat{Q}^{(i)}_{\text{target}}(s', \pi(s')|\Omega_s^{d_k})$. The actor is trained to maximize the estimated Q-value while staying close to the dataset policy:

$$\mathcal{L}_{\text{actor}}^{\texttt{TD3+BC}} = -\mathbb{E}_{s\sim\mathcal{D}}\left[\hat{Q}(s,\pi(s)|\Omega_s^{d_k})\right] + \alpha \cdot \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[\|\pi(s)-a\|^2\right]. \tag{53}$$

Experiment results can be found at Table 3.

Table 3: Evaluation for TD3+BC based ICQL variant on Mujoco and Adroit tasks. Average normalized scores are reported over 5 random seeds.

Mujoco Tasks	**TD3-BC**	**ICQL-TD3-BC(ours)**	**Gain(%)**
Walker2d-Medium-Expert-v2	109.19	**109.27**	0.07%
Walker2d-Medium-v2	**77.02**	72.67	-5.65%
Walker2d-Medium-Replay-v2	41.47	**54.96**	32.53%
Hopper-Medium-Expert-v2	78.16	**87.16**	11.51%
Hopper-Medium-v2	53.49	**57.93**	8.30%
Hopper-Medium-Replay-v2	59.36	**65.81**	10.87%
HalfCheetah-Medium-Expert-v2	62.78	**63.74**	1.53%
HalfCheetah-Medium-v2	**43.09**	42.74	-0.81%
HalfCheetah-Medium-Replay-v2	41.76	**45.86**	9.82%
Average	62.92	**66.68**	6.00%
Adroit Tasks	**TD3-BC**	**ICQL-TD3-BC(ours)**	**Gain(%)**
Pen-Human-v1	64.62	**68.29**	5.68%
Pen-Cloned-v1	**76.82**	74.71	-2.75%
Hammer-Human-v1	1.52	**1.64**	7.89%
Hammer-Cloned-v1	1.81	**7.25**	300.55%
Door-Human-v1	0.15	**2.03**	1253.33%
Door-Cloned-v1	**-0.05**	-0.08	-60.00%
Average	24.15	**25.64**	6.17%

H IMPLEMENTATION DETAILS

In this section, we present the detailed network architecture for our in-context critic and actor. In addition, we describe the hyperparameter settings in this paper.

H.1 IN-CONTEXT CRITIC NETWORK

The In-Context Critic is composed of a feature extractor and a linear transformer. The feature extractor is a 3-layer MLP with 256 hidden units. A Tanh function is applied as the last layer activation, and ReLU is applied as activation function for other layers, followed by layer normalization. The output dimension of the feature extractor is 64. A dropout rate of 0.1 is applied during training the feature extractor. The linear transformer is built as described in Equation (16), where trainable parameters exist only in G. The definition of G is in Equation (23), where C_l denotes the trainable parameters in the l-th layer. The shape of C_l is 64×64. We use gradient normalization to stabilize training by scaling the gradients to have a maximum L2 norm of 10. The number of linear transformer layers is set to 20.

H.2 POLICY NETWORK

For `ICQL-IQL`, the policy network is built as an MLP with 2 hidden layers and the ReLU activation function. The policy network contains an additional learnable vector representing the logarithmic standard deviation of actions. A dropout rate of 0.1 is applied during training.

For `ICQL-TD3+BC`, the policy network is built as a 3-layer MLP with the ReLU activation function.

H.3 HYPER-PARAMETER SETTINGS

For `ICQL-IQL`, we follow the original IQL paper and set different hyperparameter expectile τ and temperature β for different offline datasets. We searched among $\{0.5, 0.7, 0.9\}$ for expectile and $\{1, 2, 3\}$ for temperature. The detailed list is in Table 4.

Table 4: Expectile and temperature settings for `ICQL` experiments.

Tasks	Expectile	Temperature	Tasks	Expectile	Temperature
Walker2d-Medium-Expert-v2	0.7	1	Pen-Human-v1	0.7	2
Walker2d-Medium-v2	0.7	1	Pen-Cloned-v1	0.9	2
Walker2d-Medium-Replay-v2	0.7	1	Hammer-Human-v1	0.5	1
Hopper-Medium-Expert-v2	0.7	1	Hammer-Cloned-v1	0.9	2
Hopper-Medium-v2	0.5	1	Door-Human-v1	0.5	1
Hopper-Medium-Replay-v2	0.7	2	Door-Cloned-v1	0.7	2
HalfCheetah-Medium-Expert-v2	0.5	2	Kitchen-Complete-v0	0.9	1
HalfCheetah-Medium-v2	0.5	1	Kitchen-Mixed-v0	0.5	1
HalfCheetah-Medium-Replay-v2	0.7	1	Kitchen-Partial-v0	0.9	2

For `ICQL-TD3+BC`, we follow the settings of the original paper, using the same hyperparameter $\alpha = 2.5$ for all datasets.

Other common hyperparameters are listed in Table 5.

Table 5: Common hyperparameters for `ICQL` main experiments.

Hyperparameter	Value
Hidden dimension	256
Batch size	256
Training steps	1,000,000
Evaluation episodes	10
Discount factor	0.99
Policy learning rate	3.0e-4
Critic learning rate	3.0e-4
Context length	20

H.4 Retrieval Strategies

In Section 4.3, we have compared the performance of ICQL while using different strategies for retrieving context for approximating the localized Q function. The description of retrieval strategies in Section 4.3 are as follows:

- **State-Similar Retrieval**: Given current state s, search for 20 similar states s_i from the offline dataset using cosine similarity, and retrieve their corresponding transitions $\{s_i, a_i, r_i, s'_i, a'_i\}$.
- **Random Retrieval**: Given current state s, randomly select 20 transitions $\{s_i, a_i, r_i, s'_i, a'_i\}$ as context.
- **State-Similar-with-High-Rewards**: Given current state s, search for 60 similar states s_i from the offline dataset using cosine similarity, retrieve their corresponding transitions $\{s_i, a_i, r_i, s'_i, a'_i\}$. Then sort by the rewards r_i in these retrieved transitions, and select 20 transitions with the highest rewards as context.

H.5 Analysis on In-Context Critics

In this section, we conduct further analysis into the functionality of our in-context Q estimator. By construction, the forward pass of our in-context Q estimator is equivalent to the step-wise optimization of TD-error. We analyze the outputs and the parameter distributions of each intermediate layer to validate its effectiveness. We randomly select 10 different states and their corresponding action in the offline dataset of Walker2d-Medium-Expert-v2, retrieve 20 relevant transitions by best cosine state similarity, and estimate the Qs for these state-action pairs. We store outputs of all intermediate layers and the visualization results are shown in Figure 6. From Figure 6 we can discover that the Q estimates show converging trend as the layer get deeper, validating the iterative refinement process.

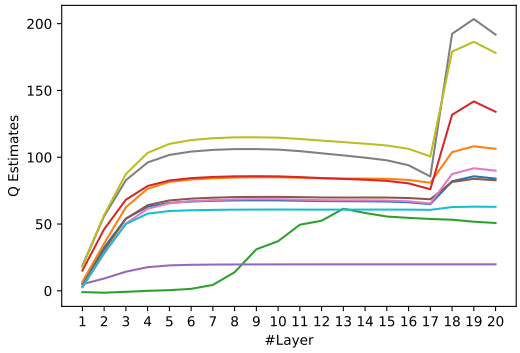


Figure 6: Q-estimates of each intermediate layers.

I Additional Experiment Results

I.1 Extended baselines

In this section, we extend our comparisons with the more methods (RA-DT Schmied et al. (2024), ReBRAC Tarasov et al. (2023), DMG Mao et al. (2024), FQL Park et al. (2025), QC Li et al. (2025)), following their official implementations. ICQL demonstrates competitive or superior performance across most tasks. Results are shown in Table 6.

I.2 Numerical results for ablation studies on the number of layers and context lengths

In this section, we provide numerical results in correspondence to Section 4.3.1 and Section 4.3.2.

Table 6: Performance comparison across Mujoco, Adroit, and Kitchen tasks. Average and standard deviation of scores are reported over 5 random seeds.

Task	BC	TD3BC	CQL	IQL	DT	RADT	ReBRAC	DMG	FQL	QC	ICQL
Walker2d-ME	107.5	109.2	98.7	109.8	70.7	107.8	109.2	109.5	101.0	102.8	**113.3**
Walker2d-M	75.3	77.0	79.2	71.5	70.2	68.9	82.8	**85.0**	72.4	34.1	80.3
Walker2d-MR	26.0	41.5	77.2	61.0	54.8	67.2	39.4	81.9	60.9	46.6	**81.9**
Hopper-ME	52.5	78.2	105.4	98.5	57.5	109.4	98.7	109.8	60.1	44.0	**113.3**
Hopper-M	52.9	53.5	58.0	63.3	57.1	62.4	60.6	**92.3**	55.6	64.6	62.6
Hopper-MR	18.1	59.4	95.0	82.4	65.8	81.6	87.4	**100.1**	55.0	18.6	96.4
HalfCheetah-ME	55.2	62.8	62.4	83.4	70.8	90.9	84.6	93.6	92.9	**94.2**	89.1
HalfCheetah-M	42.6	43.1	44.4	42.5	42.8	42.0	44.6	47.9	43.9	**48.2**	45.9
HalfCheetah-MR	36.6	41.8	**45.5**	38.9	39.5	38.9	36.9	44.6	40.0	40.5	44.7
Pen-Human	63.9	64.6	37.5	89.5	79.5	17.8	**91.5**	66.2	61.2	55.7	85.6
Pen-Cloned	37.0	76.8	39.2	84.9	74.0	32.4	68.9	67.5	23.5	54.8	**89.4**
Hammer-Human	1.2	1.5	4.4	7.2	1.7	0.7	1.1	**18.4**	1.1	1.2	3.7
Hammer-Cloned	0.6	1.8	2.1	0.5	3.7	1.3	0.2	**13.4**	1.7	2.2	4.5
Door-Human	2.0	0.2	9.9	9.8	5.5	13.2	-0.1	0.1	0.2	0.7	**17.1**
Door-Cloned	0.0	-0.1	0.1	7.6	3.2	2.4	9.0	3.7	0.1	4.4	**11.7**
Kitchen-Complete	65.0	57.5	43.8	59.2	52.5	32.5	60.0	22.5	16.3	27.5	**79.3**
Kitchen-Mixed	51.5	53.5	51.0	53.3	**60.0**	54.1	47.5	30.0	45.0	**60.0**	59.5
Kitchen-Partial	38.0	46.7	49.8	45.8	55.0	53.8	62.5	37.5	15.8	52.5	**61.5**
Overall Average	47.3	47.7	58.5	63.2	56.2	57.2	60.0	62.0	50.5	47.7	**69.7**

Table 7: Normalized scores for Gym tasks with different lengths of contexts and different number of layers in `ICQL-IQL`.

	Context Length				Number of Layers			
Gym Tasks	**10**	**20**	**30**	**40**	**4**	**8**	**16**	20
Walker2d-Medium-Expert-v2	111.07	**113.23**	111.71	110.18	102.27	103.28	104.06	**113.23**
Walker2d-Medium-v2	79.59	79.59	70.9	**80.68**	78.04	78.35	74.93	**79.59**
Walker2d-Medium-Replay-v2	77.46	**84.81**	69.43	74.38	76.27	76.97	75.78	**84.81**
Hopper-Medium-Expert-v2	103.68	**110.67**	105.99	103.42	104.76	**111.78**	106.96	110.67
Hopper-Medium-v2	**73.82**	67.36	60.18	59.43	65.65	**67.62**	67.3	67.36
Hopper-Medium-Replay-v2	89.89	**91.63**	81.21	83.92	**100.53**	97.84	91.77	91.63
HalfCheetah-Medium-Expert-v2	89.23	**90.3**	88.76	83.48	71.29	63.31	74.84	**90.3**
HalfCheetah-Medium-v2	45.85	46.08	**46.28**	45.82	45.05	44.77	45.01	**46.08**
HalfCheetah-Medium-Replay-v2	43.7	**44.48**	44.29	44.19	43.5	43.64	43.75	**44.48**
Average	79.37	**80.91**	75.42	76.17	76.37	76.40	76.04	**80.91**

I.3 Comparison on different in-context modeling choices

We performed additional experiments replacing the linear transformer with other architectures, which is either a small MLP or a standard transformer. The results are shown in Table 8. The results demonstrate that the linear in-context mechanism is not only theoretically convenient for but also empirically essential for learning local Q function.

Table 8: Performance comparison across different local modeling choices: linear attention, linear MLP, and standard self-attention.

Task	Linear Transformer	Linear MLP	Standard Transformer
Walker2d-Medium-Expert	**113.3**	109.5	108.8
Walker2d-Medium	**80.3**	76.7	77.4
Walker2d-Medium-Replay	**81.9**	60.2	42.9
Hopper-Medium-Expert	**113.3**	109.9	70.3
Hopper-Medium	**62.6**	55.7	61.9
Hopper-Medium-Replay	**96.4**	89.9	42.1
HalfCheetah-Medium-Expert	**89.1**	83.0	72.5
HalfCheetah-Medium	**45.9**	43.3	42.0
HalfCheetah-Medium-Replay	**44.7**	39.2	36.1
Pen-Human	**85.6**	66.6	72.7
Pen-Clone	**89.4**	80.7	83.8
Hammer-Human	3.7	**6.1**	4.2
Hammer-Clone	4.5	**7.9**	1.8
Door-Human	**17.1**	6.9	8.9
Door-Cloned	**11.7**	3.5	3.4
Kitchen-Complete	**79.3**	70.0	78.3
Kitchen-Mixed	**59.5**	57.5	55.8
Kitchen-Partial	**61.5**	48.3	55.8

I.4 Computation overhead analysis

I.4.1 Comparison on training time, inference time, GFLOPs and memory consumption

In this section, we compare training time, inference time, GFLOPs and memory consumption across all baseline methods. The analysis is conducted on Walker2d-Medium-Expert dataset, and the results are summarized in Table 9. This analysis shows that while ICQL incurs moderate additional compute cost relative to most advanced baselines, and it remains more efficient than sequential models (DT/RADT) while achieving substantially stronger performance.

Table 9: Computation cost comparison across offline RL algorithms, including per-step training/inference time, FLOPs, and peak memory consumption.

Algorithm	Train Time (ms)	Infer Time (ms)	Training GFLOPs	Peak Memory (MB)
TD3BC	7.23	0.26	0.17	30
IQL	10.52	0.61	0.22	26
CQL	47.57	0.61	2.64	79
DT	68.42	2.89	151.40	1383
RA-DT	121.02	3.13	1103.79	1424
ReBRAC	13.91	0.26	0.18	38
DMG	32.33	0.42	0.55	27
FQL	19.63	0.37	4.53	126
QC	21.60	0.25	4.65	244
ICQL	70.73	0.51	1.03	375

I.4.2 Analysis on GFLOPs and memory consumption scaling of ICQL

We further report training GFLOPs and memory consumption for varying context lengths in $\{10, 20, 30, 40\}$ and varing number of linear transformer layers, in Table 10 and Table 11. The training time needed scales with both context length and number of layers. Using a context length of 20 and 20 linear transformer layers remains comparable efficient while providing competitive performance.

Table 10: Training FLOPs (in GFLOPs) for different numbers of layers and context lengths K.

# Layers	K=10	K=20	K=30	K=40
10	0.25	0.51	0.81	1.14
20	0.50	1.03	1.62	2.28
30	0.75	1.54	2.43	3.42
40	1.00	2.06	3.24	4.56

Table 11: Peak memory consumption (in MB) for different numbers of layers and context lengths K.

# Layers	K=10	K=20	K=30	K=40
10	171.28	306.56	445.57	590.39
20	209.71	375.38	549.51	738.00
30	248.39	443.29	655.26	879.30
40	288.94	511.58	758.94	**1023.75**

I.4.3 Detailed comparison on retrieval and training time of ICQL across all datasets

To mitigate repeated computation, we pre-compute all retrieval indices once before training, since: 1) The offline dataset is fixed. 2) The retrieval rule is deterministic. 3) Pre-computation does not affect the learning dynamics or outcomes. This turns per-step retrieval cost into an amortized constant-time lookup during training. ICQL follows a standard actor–critic-like training paradigm where the critic uses retrieved context to estimate local Q-values and the policy learns from these Q-values. At evaluation time, only the policy is used, which is consistent with standard actor-critic practice. We report the real-time retrieval time, the lookup time with cached indices, and the training/inference speed for all datasets. The results are averaged across all datasets used in our experiments. As shown in Table 12, cached retrieval adds only 0.03 ms per step, which is negligible relative to the overall training time. The breakdown analysis of retrieval time and training/inference time analysis are provided in Table 13 and Table 14.

Table 12: Average ICQL runtime of retrieval, training with different context lengths, and inference, across all datasets.

	Time (ms)
Retrieval with Cached Index	0.03
Train with K=10	46.94
Train with K=20	72.15
Train with K=30	113.86
Train with K=40	171.95
Inference	0.54

Table 13: Detailed retrieval time (ms) analysis across tasks and context lengths. Cached index retrieval eliminates repeated nearest-neighbor searches and greatly reduces overhead.

Task	Dataset Size	K=10	K=20	K=30	K=40	Cached
Walker2d-Medium-Expert	1998318	6.38	6.52	6.90	7.70	0.04
Walker2d-Medium	999322	3.98	3.96	4.37	5.16	0.03
Walker2d-Medium-Replay	301698	1.92	2.18	2.64	3.90	0.03
Hopper-Medium-Expert	1998966	6.04	6.11	6.39	7.31	0.03
Hopper-Medium	999998	3.85	3.71	4.05	4.77	0.03
Hopper-Medium-Replay	401598	2.10	2.16	2.56	3.11	0.03
HalfCheetah-Medium-Expert	1998000	6.27	6.41	6.75	7.37	0.03
HalfCheetah-Medium	999000	3.96	3.90	4.24	5.17	0.04
HalfCheetah-Medium-Replay	201798	1.58	1.61	1.81	2.53	0.03
Pen-Human	4975	0.89	0.81	0.99	1.15	0.03
Pen-Cloned	496264	2.91	3.05	3.52	6.67	0.03
Hammer-Human	11285	0.88	0.89	1.05	1.17	0.03
Hammer-Cloned	996394	4.56	4.54	4.93	5.82	0.03
Door-Human	6704	0.88	0.88	1.07	1.17	0.03
Door-Cloned	995642	4.39	4.53	4.92	5.94	0.03
Kitchen-Complete	3679	0.89	0.82	0.95	1.12	0.03
Kitchen-Partial	136937	1.36	1.41	1.60	1.91	0.03
Kitchen-Mixed	136937	1.49	1.38	1.63	2.11	0.03

Table 14: Training and inference time (ms) for different context lengths across tasks. Training time grows approximately linearly with the context length, while inference time remains nearly constant.

Task	K=10	K=20	K=30	K=40	Inference
Walker2d-Medium-Expert	48.90	70.73	111.75	170.71	0.51
Walker2d-Medium	46.63	71.77	113.82	170.85	0.50
Walker2d-Medium-Replay	48.68	74.75	115.43	171.97	0.52
Hopper-Medium-Expert	48.31	70.71	114.56	173.52	0.51
Hopper-Medium	46.39	71.60	113.35	171.63	0.57
Hopper-Medium-Replay	46.08	72.32	112.89	170.58	0.56
HalfCheetah-Medium-Expert	48.33	73.27	115.90	171.46	0.58
HalfCheetah-Medium	47.69	74.45	113.85	171.75	0.51
HalfCheetah-Medium-Replay	47.30	71.81	114.32	172.86	0.57
Pen-Human	45.65	72.41	114.23	171.73	0.56
Pen-Cloned	44.50	69.59	112.02	170.31	0.51
Hammer-Human	46.88	73.78	113.93	171.66	0.52
Hammer-Cloned	47.31	72.55	114.13	171.94	0.57
Door-Human	46.16	71.34	113.11	171.44	0.57
Door-Cloned	45.37	71.20	112.11	170.61	0.58
Kitchen-Complete	47.45	73.65	116.36	175.98	0.54
Kitchen-Partial	48.11	72.35	116.50	175.42	0.52
Kitchen-Mixed	45.25	70.47	111.17	170.59	0.52

I.5 Failure Analysis on Hammer dataset

In this section, we provide a failure analysis on Hammer-Human dataset. We found that Hammer-Human exhibits two properties that make it particularly challenging for ICQL:

1) Small dataset size and sparse coverage. Hammer-Human contains only 24 trajectories (~11k transitions), vastly fewer than Hammer-Cloned (~996k transitions). This leads to large distances between the query state and its retrieved neighbors that violate locality assumptions, and poor state-space coverage that make retrieval more likely to pull in semantically irrelevant transitions.

2) Low-quality transitions and noisy rewards. Most Hammer-Human trajectories have very low returns. So for each query state, the retrieved neighbors tend to have weak reward signals, making it more difficult to fit effective local Q-function.

We provide dataset statistics comparisons in Table 15, and comparison of distributions of mean distance between query states and retrieved states in Figure 7, both of which confirm our observations.

Table 15: Dataset statistics for Hammer-Human and Hammer-Cloned.

Dataset	**Hammer-Human**	**Hammer-Cloned**
Number of trajectories	24	3605
Number of transitions	11285	996394
Mean Trajectory Length (Min–Max)	455.2 (347–623)	276.4 (199–623)
Mean Trajectory Return (Min–Max)	2817.5 (-109–16022)	779.8 (-407–16022)

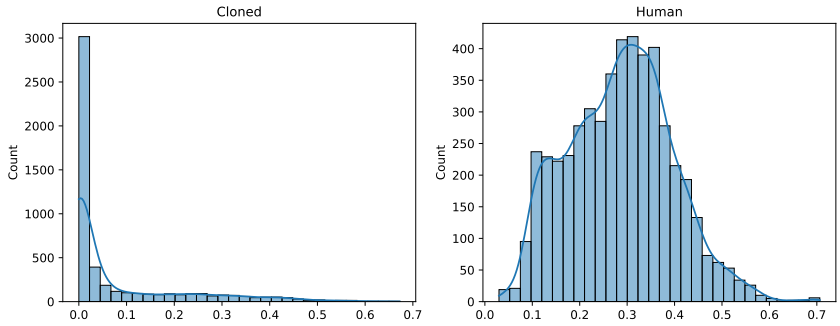


Figure 7: Distribution of mean distance between query states and retrieved states on Hammer dataset.

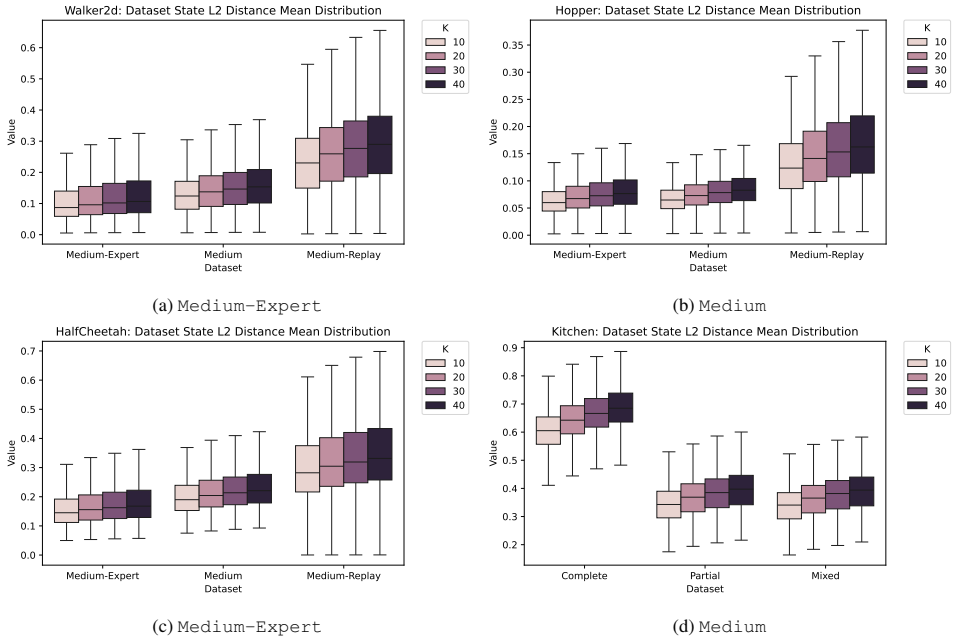
Although Hammer-Cloned contains mostly low-return behavior, the large dataset size provides much denser coverage. ICQL can retrieve states that are substantially closer to the query state, enabling more reliable local linear approximation and producing slightly higher scores.

Moreover, we would like to note that for both Hammer-Human and Hammer-Cloned, the extremely low proportion of high-reward transitions makes it inherently difficult to retrieve any local neighborhood that provides strong positive supervision. As a result, even if the Q-network successfully fits a local linear approximation, it rarely observes transitions that reliably correspond to high-return behavior. Consequently, the learned Q-values cannot meaningfully distinguish truly rewarding actions, leading to uniformly low evaluation scores across both datasets.

I.6 Analysis on the relationship among theoretical d and K

In the theory, a local set $\Omega^d_{s_{query}}$ is defined as all transitions whose states fall within a radius-d neighborhood around s. This radius determines the intrinsic "locality scale" at which the Q-function is assumed to be approximately linear. However, in practice, the radius d is not directly tunable: it depends on the underlying density and geometry of the dataset and is unknown to the algorithm.

Instead, `ICQL` controls locality through the retrieval size k. Retrieving the top-k nearest neighbors is equivalent to selecting a data-adaptive radius, where $d_k = \max_{(s_i,\cdot)\in\text{top-}k} ||s_i - s||_2^2$ and $\bar{d}_k = \max_{(s_i,\cdot)\in\text{top-}k} ||s'_i - s'||_2^2$, so that the practical neighborhood is exactly the theoretical local set with radius $(d_k, \bar{d}_k)$. The distribution of mean distance between query states and retrieved states of different k is visualized in Figure 8.



(a) `Medium-Expert` (b) `Medium`

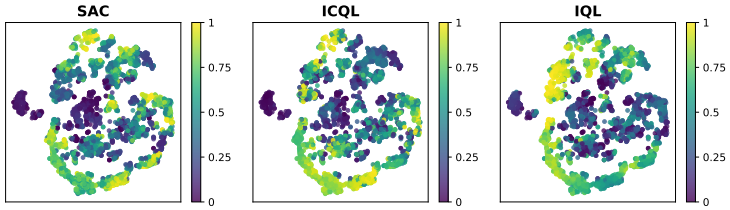
(c) `Medium-Expert` (d) `Medium`

Figure 8: Distribution of mean distance between query states and retrieved states of different k.

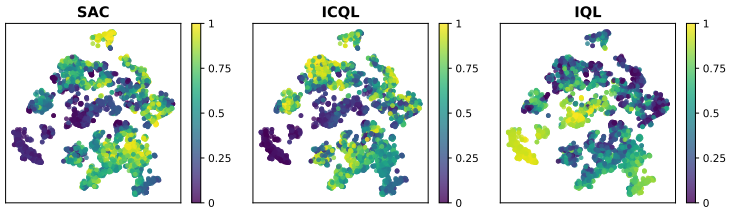
Thus, k determines the effective radius implicitly and monotonically: larger k expands the radius $(d_k, \bar{d}_k)$ and increases the size and heterogeneity of $\Omega^{d_k}_s$, while smaller k leads to tighter neighborhoods with more consistent local value structure.

I.7 ADDITIONAL VISUALIZATION ON LEARNED Q-VALUE COMPARISON

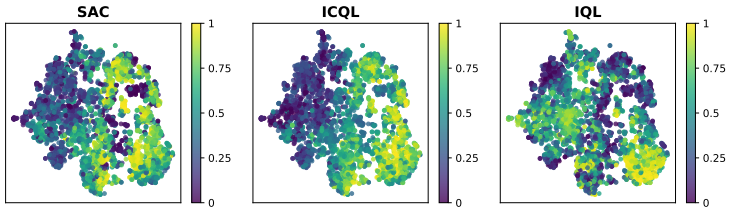
We extend the visualization analysis of learned Q-values of ICQL and IQL by comparing with Q-value learned with online RL method SAC on Walker2d-Medium-Expert, Walker2d-Medium and Walke2d-Medium-Replay datasets. We scale all Q estimates into the [0,1] range before visualization. We also include additional scatter plots comparing each method's estimated Q-values against the SAC oracle. The visualization are shown in Figure 9 and Figure 10. These plots clearly show that the correlation patterns between ICQL and SAC is better than that between IQL and SAC, indicating ICQL can produce more accurate value estimation than IQL.



(a) Medium-Expert

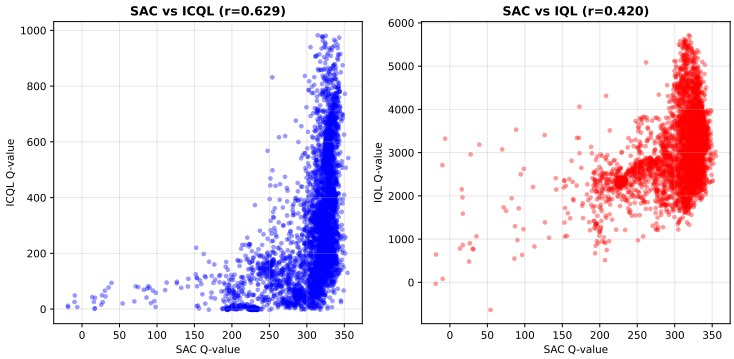


(b) Medium

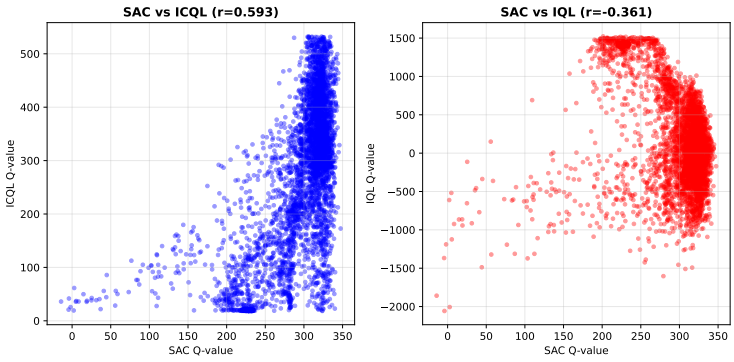


(c) Medium-Replay

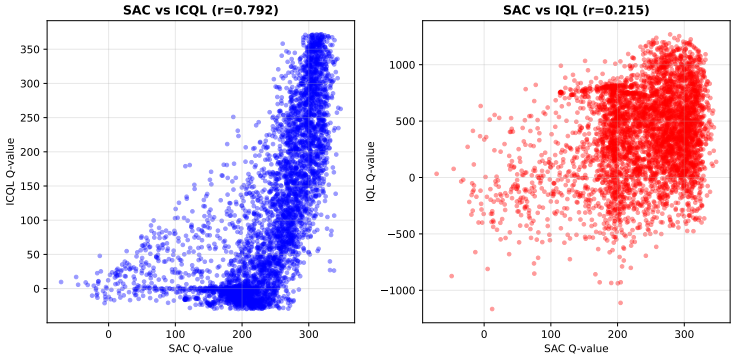
Figure 9: Q-value of Walker2d-Medium-Expert, Walker2d-Medium, and Walker2d-Medium-Replay dataset on t-SNE mapped state distribution.



(a) `Medium-Expert`



(b) `Medium`



(c) `Medium-Replay`

Figure 10: Q-value correlation of Walker2d-Medium-Expert, Walker2d-Medium, and Walker2d-Medium-Replay dataset. Red: Correlation between Q-values learned by `IQL` and `SAC`. Blue: Correlation between Q-values learned by `ICQL` and `SAC`.