MODEL-BASED OFFLINE REINFORCEMENT LEARNING WITH LOWER EXPECTILE Q-LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Model-based offline reinforcement learning (RL) is a compelling approach that addresses the challenge of learning from limited, static data by generating imaginary trajectories using learned models. However, these approaches often struggle with inaccurate value estimation from model rollouts. In this paper, we introduce a novel model-based offline RL method, Lower Expectile Q-learning (LEQ), which provides a low-bias model-based value estimation via lower expectile regression of λ-returns. Our empirical results show that LEQ significantly outperforms previous model-based offline RL methods on long-horizon tasks, such as the D4RL AntMaze tasks, matching or surpassing the performance of model-free approaches and sequence modeling approaches. Furthermore, LEQ matches the performance of state-of-the-art model-based and model-free methods in dense-reward environments across both state-based tasks (NeoRL and D4RL) and pixel-based tasks (V-D4RL), showing that LEQ works robustly across diverse domains. Our ablation studies demonstrate that lower expectile regression, λ-returns, and critic training on offline data are all crucial for LEQ.

1 INTRODUCTION

One of the major challenges in offline reinforcement learning (RL) is the overestimation of values for out-of-distribution actions due to the lack of environment interactions (Levine et al., 2020; Kumar et al., 2020). Model-based offline RL addresses this issue by generating additional (imaginary) training data using a learned model, thereby augmenting the given offline data with synthetic experiences that cover out-of-distribution states and actions (Yu et al., 2020; Kidambi et al., 2020; Yu et al., 2021; Argenson and Dulac-Arnold, 2021; Sun et al., 2023). While these approaches have demonstrated strong performance in simple, short-horizon tasks, they struggle with noisy model predictions and value estimates, particularly in long-horizon tasks (Park et al., 2024). This challenge is evident in their poor performances (i.e. near zero) on the D4RL AntMaze tasks (Fu et al., 2020).

Typical model-based offline RL methods alleviate the inaccurate value estimation problem (mostly overestimation) by penalizing Q-values estimated from model rollouts with uncertainties in model predictions (Yu et al., 2020; Kidambi et al., 2020) or value predictions (Sun et al., 2023; Jeong et al., 2023). While these penalization terms prevent a policy from exploiting erroneous value estimates, the policy now does not maximize the true value, but maximizes the value penalized by heuristically estimated uncertainties, which can lead to sub-optimal behaviors. This is especially problematic in long-horizon, sparse-reward tasks, where Q-values are similar across nearby states (Park et al., 2024).

Another way to reduce bias in value estimates is using multi-step returns (Sutton, 1988; Hessel et al., 2018). CBOP (Jeong et al., 2023) constructs an explicit distribution of multi-step Q-values from thousands of model rollouts and uses this value as a target for training the Q-function. However, CBOP is computationally expensive for estimating a target value and uses multi-step returns solely for Q-learning, not for policy optimization.

To tackle these issues in model-based offline RL, we introduce a simple yet effective model-based offline RL algorithm, Lower Expectile Q-learning (LEQ). As illustrated in Figure 1, LEQ uses expectile regression with a small τ for both policy and Q-function training, providing an efficient and elegant way to achieve conservative Q-value estimates. Moreover, we propose to optimize both

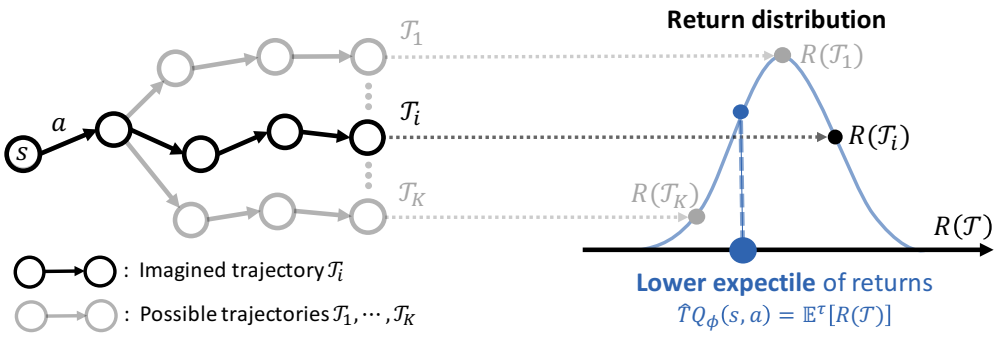


Figure 1: **Lower Expectile Q-learning (LEQ). (left)** In model-based offline RL, an agent can generate imaginary trajectories using a world model. **(right)** For conservative Q-evaluation of the policy, LEQ learns the **lower expectile** of the target Q-distribution from a few sampled rollouts $\mathcal{T}_i$, without estimating the entire Q-distribution with exhaustive rollouts.

policy and Q-function using λ-returns (i.e. TD(λ) targets) of long (10-step) model rollouts, allowing the policy to directly learn from low-bias multi-step returns (Schulman et al., 2016).

The experiments on the D4RL AntMaze, MuJoCo Gym (Fu et al., 2020), NeoRL (Qin et al., 2022), and V-D4RL (Lu et al., 2023) benchmarks show that LEQ improves model-based offline RL across diverse domains. To the best of our knowledge, LEQ is the first model-based offline RL algorithm capable of outperforming model-free offline RL algorithms on the long-horizon AntMaze tasks (Fu et al., 2020; Jiang et al., 2023). Moreover, LEQ matches the top scores across various benchmarks, while prior methods demonstrate superior performances only for a specific domain.

2 Related Work

Offline RL (Levine et al., 2020) aims to solve an RL problem only with pre-collected datasets, outperforming behavioral cloning policies (Pomerleau, 1989). While it is possible to apply off-policy RL algorithms on fixed datasets, these algorithms suffer from the overestimation of Q-values for actions unseen in the offline dataset (Fujimoto et al., 2019; Kumar et al., 2019; 2020) since the overestimated values cannot be corrected through interactions with environments as in online RL.

Model-free offline RL algorithms have addressed this value overestimation problem on out-of-distribution actions by (1) regularizing a policy to only output actions in the offline data (Peng et al., 2019; Kostrikov et al., 2022; Fujimoto and Gu, 2021) or (2) adopting a conservative value estimation for executing actions different from the dataset (Kumar et al., 2020; An et al., 2021). Despite their strong performances on the standard offline RL benchmarks, model-free offline RL policies tend to be constrained to the support of the data (i.e. state-action pairs in the offline dataset), which may lead to limited generalization capability.

Model-based offline RL approaches have tried to overcome this limitation by suggesting a better use of the limited offline data – learning a world model and generating imaginary data with the learned model that covers out-of-distribution actions. Similar to Dyna-style online model-based RL (Sutton, 1991; Hafner et al., 2019; 2021; 2023), an offline model-based RL policy can be trained on both offline data and model rollouts. But, again, learned models may be inaccurate on states and actions outside the data support, making a policy easily exploit the learned models.

Recent model-based offline RL algorithms have adopted the conservatism idea from model-free offline RL, penalizing policies incurring (1) uncertain transition dynamics (Yu et al., 2020; Kidambi et al., 2020; Yu et al., 2021) or (2) uncertain value estimation (Sun et al., 2023; Jeong et al., 2023). This conservative use of model-generated data enables model-based offline RL to outperform model-free offline RL in widely used offline RL benchmarks (Sun et al., 2023). However, uncertainty estimation is difficult and often inaccurate (Yu et al., 2021). Instead of relying on such heuristic (Yu et al., 2020; Kidambi et al., 2020; Sun et al., 2023) or expensive (Jeong et al., 2023) uncertainty

Q-learning		**Lower Expectile Q-learning**
$\mathcal{L}_Q(\phi) = (Q_\phi(\mathbf{s}_t, \mathbf{a}_t) - Q_t^\lambda(\mathcal{T}))^2$	$\Longrightarrow$	$\lvert\tau - \mathbb{1}(Q_t^\lambda(\mathcal{T}) > Q_\phi(\mathbf{s}_t, \mathbf{a}_t))\rvert \cdot \mathcal{L}_Q(\phi)$
$\nabla_\theta \mathcal{L}_\pi(\theta) = -\nabla_\theta \pi_\theta(\mathbf{s}_t) \cdot \nabla_{\mathbf{a}_t} Q_t^\lambda(\mathcal{T})$	$\Longrightarrow$	$\lvert\tau - \mathbb{1}(Q_t^\lambda(\mathcal{T}) > Q_\phi(\mathbf{s}_t, \mathbf{a}_t))\rvert \cdot \nabla_\theta \mathcal{L}_\pi(\theta)$

Figure 2: **Comparison of standard Q-learning and Lower Expectile Q-learning (LEQ).** LEQ generalizes standard Q-learning (with λ-returns $Q_t^\lambda(\mathcal{T})$) by multiplying a simple asymmetric weight "$|\tau - \mathbb{1}(Q_t^\lambda(\mathcal{T}) > Q_\phi(s_t, a_t))|$" to the Q-learning objectives. $\mathcal{T} = (\mathbf{s}_0, \mathbf{a}_0, r_0, \mathbf{s}_1, \mathbf{a}_1, r_1, \cdots, \mathbf{s}_T)$ is a model-generated trajectory and $\tau \leq 0.5$ is the expectile hyperparameter controlling the degree of conservatism. When $\tau = 0.5$, LEQ reduces to standard Q-learning.

estimation, we propose to learn a conservative value function via expectile regression with a small τ, which is simple, efficient, yet effective, as illustrated in Figure 2.

Expectile regression for offline RL has been first introduced by IQL (Kostrikov et al., 2022), which has been extended to model-based offline RL, such as IQL-TD-MPC (Chitnis et al., 2024). IQL uses *"upper expectile"* to approximate the *"max operation"* in $V(s) = \max_a Q(s, a)$ without querying out-of-distribution actions. On the other hand, our work **fundamentally differs from IQL-like approaches** in that our method uses *"lower expectile"* to get *"conservative return estimates"* from trajectories generated by potentially inaccurate model rollouts.

3 Preliminaries

Problem setup. We formulate our problem as a Markov Decision Process (MDP) defined as a tuple, $\mathcal{M} = (\mathcal{S}, \mathcal{A}, r, p, \rho, \gamma)$ (Sutton and Barto, 2018). $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, respectively. $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ denotes the reward function. $p : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$[1] denotes the transition dynamics. $\rho(\mathbf{s}_0) \in \Delta(\mathcal{S})$ denotes the initial state distribution and γ is a discounting factor. The goal of RL is to find a policy, $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, that maximizes the expected return, $\mathbb{E}_{\mathcal{T} \sim p(\cdot \mid \pi, \mathbf{s}_0 \sim \rho)} \left[\sum_{t=0}^{T-1} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t) \right]$, where $\mathcal{T}$ is a sequence of transitions with a finite horizon T, $\mathcal{T} = (\mathbf{s}_0, \mathbf{a}_0, r_0, \mathbf{s}_1, \mathbf{a}_1, r_1, ..., \mathbf{s}_T)$, following $\pi(\mathbf{a}_t \mid \mathbf{s}_t)$ and $p(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)$ starting from $\mathbf{s}_0 \sim \rho(\cdot)$.

In this paper, we consider the offline RL setup (Levine et al., 2020), where a policy π is trained with a fixed given offline dataset, $\mathcal{D}_{\text{env}} = \{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_N\}$, without any additional online interactions.

Model-based offline RL. As an offline RL policy is trained from a fixed dataset, one of the major challenges in offline RL is the limited data support; thus, lack of generalization to out-of-distribution states and actions. Model-based offline RL (Kidambi et al., 2020; Yu et al., 2020; 2021; Rigter et al., 2022; Sun et al., 2023; Jeong et al., 2023) tackles this problem by augmenting the training data with imaginary training data (i.e. model rollouts) generated from the learned transition dynamics and reward model, $p_\psi(\mathbf{s}_{t+1}, r_t \mid \mathbf{s}_t, \mathbf{a}_t)$.

The typical process of model-based offline RL is as follows: (1) pretrain a model (or an ensemble of models) and an initial policy from the offline data, (2) generate short imaginary rollouts $\{\mathcal{T}\}$ using the pretrained model and add them to the training dataset $\mathcal{D}_{\text{model}} \leftarrow \mathcal{D}_{\text{model}} \cup \{\mathcal{T}\}$, (3) perform an offline RL algorithm on the augmented dataset $\mathcal{D}_{\text{model}} \cup \mathcal{D}_{\text{env}}$, and repeat (2) and (3).

Expectile regression. Expectile is a generalization of the expectation of a distribution X. While the expectation of X, $\mathbb{E}[X]$, can be viewed as a minimizer of the least-square objective, $\mathbb{E}_{x \sim X}[L_2(y - x)] = \mathbb{E}_{x \sim X}[\frac{1}{2}(y - x)^2]$, τ-expectile of X, $\mathbb{E}^\tau[X]$, can be defined as a minimizer of the asymmetric least-square objective $\mathbb{E}_{x \sim X}[L_2^\tau(y - x)]$, where $L_2^\tau(\cdot)$ is defined as:

$$L_2^\tau(u) = |\tau - \mathbb{1}(u > 0)| \cdot u^2, \tag{1}$$

where $|\tau - \mathbb{1}(u > 0)|$ is an asymmetric weighting of least-squared objective with $0 \leq \tau \leq 1$.

We refer to a τ-expectile with $\tau < 0.5$ as a *lower expectile* of X. When $\tau < 0.5$, the objective assigns a high weight $1 - \tau$ for smaller x and a low weight τ for bigger x. Thus, minimizing the objective with $\tau < 0.5$ leads to a *conservative* statistical estimate compared to the expectation.

[1] $\Delta(\mathcal{X})$ denotes the set of probability distributions over $\mathcal{X}$.

Algorithm 1 LEQ: Lower Expectile Q-learning with λ-returns

Input: Offline dataset $\mathcal{D}_{\text{env}}$, expectile $\tau \leq 0.5$, imagination length H, dataset expansion length R.

Initialize world models $\{p_{\psi_1}, \cdots, p_{\psi_M}\}$, policy π_θ, and Q-function Q_ϕ
Pretrain $\{p_{\psi_1}, \cdots, p_{\psi_M}\}$ on $\mathcal{D}_{\text{env}}$ ▷ $\mathcal{L}_{\text{wm}}(\psi) = -\mathbb{E}_{(\mathbf{s},\mathbf{a},r,\mathbf{s}')\in\mathcal{D}_{\text{env}}} \log p_\psi(\mathbf{s}', r \mid \mathbf{s}, \mathbf{a})$
Pretrain π_θ and Q_ϕ on $\mathcal{D}_{\text{env}}$ ▷ using BC for π_θ and FQE (Le et al., 2019) for Q_ϕ
$\mathcal{D}_{\text{model}} \leftarrow \emptyset$
while not converged **do**
  // Expand dataset using model rollouts
  $\mathbf{s}_0 \sim \mathcal{D}_{\text{env}}$ ▷ start dataset expansion from **any** state in $\mathcal{D}_{\text{env}}$
  for $t = 0, \ldots, R-1$ **do**
    $\mathcal{D}_{\text{model}} \leftarrow \mathcal{D}_{\text{model}} \cup \{\mathbf{s}_t\}$
    $\mathbf{a}_t = \pi_\theta(\mathbf{s}_t)$
    $\mathbf{s}_{t+1}, r_t \sim p_\psi(\cdot \mid \mathbf{s}_t, \mathbf{a}_t)$, where $p_\psi \sim \{p_{\psi_1}, \cdots, p_{\psi_M}\}$
  // Generate imaginary data, $\mathcal{T} = \{(\mathbf{s}_0, \mathbf{a}_0, r_0, \cdots, \mathbf{s}_{H-1}, \mathbf{a}_{H-1}, r_{H-1}, \mathbf{s}_H)_i\}$
  $\mathbf{s}_0 \sim \mathcal{D}_{\text{model}}$ ▷ start imaginary rollout from **any** state in $\mathcal{D}_{\text{model}}$
  for $t = 0, \ldots, H-1$ **do**
    $\mathbf{a}_t = \pi_\theta(\mathbf{s}_t)$
    $\mathbf{s}_{t+1}, r_t \sim p_\psi(\cdot \mid \mathbf{s}_t, \mathbf{a}_t)$, where $p_\psi \sim \{p_{\psi_1}, \cdots, p_{\psi_M}\}$
  // Update critic using both ***model-generated data*** *and* ***offline data***
  Update critic Q_ϕ to minimize $\mathcal{L}_Q^\lambda(\phi)$ in Eq. (7) using $\mathcal{T}$ and $\{\mathbf{s}, \mathbf{a}, r, \mathbf{s}'\} \sim \mathcal{D}_{\text{env}}$
  // Update actor using only ***model-generated data***
  Update actor π_θ to minimize $\hat{\mathcal{L}}_\pi^\lambda(\theta)$ in Eq. (11) using $\mathcal{T}$

4 Approach

The primary challenge of model-based offline RL is inherent errors in a world model and critic outside the support of offline data. Conservative value estimation can effectively handle such (falsely optimistic) errors. In this paper, we introduce Lower Expectile Q-learning (LEQ), an efficient model-based offline RL method that achieves conservative value estimation via expectile regression of Q-values with lower expectiles when learning from model-generated data (Section 4.1). Additionally, we address the noisy value estimation problem (Park et al., 2024) using λ-returns on 10-step imaginary rollouts (Section 4.2). Finally, we train a deterministic policy conservatively by maximizing the lower expectile of λ-returns (Section 4.3). The overview of LEQ is described in Algorithm 1.

4.1 Lower expectile Q-learning

Most offline RL algorithms primarily focus on learning a conservative value function for out-of-distribution actions. In this paper, we propose Lower Expectile Q-learning (LEQ), which learns a conservative Q-function via expectile regression with small τ, avoiding unreliable uncertainty estimation and exhaustive Q-value estimation.

As illustrated in Figure 1, the target value for $Q_\phi(\mathbf{s}, \mathbf{a})$, where $\mathbf{a} \leftarrow \pi_\theta(\mathbf{s})$, can be estimated by rolling out an ensemble of world models and averaging $r + \gamma Q_\phi(\mathbf{s}', \mathbf{a}')$ over all possible $\mathbf{s}'$:

$$\hat{y}_{\text{model}} = \mathbb{E}_{\psi\sim\{\psi_1,\ldots,\psi_M\}} \mathbb{E}_{(\mathbf{s}',r)\sim p_\psi(\cdot|\mathbf{s},\mathbf{a})} \left[r + \gamma Q_\phi(\mathbf{s}', \pi_\theta(\mathbf{s}')) \right]. \tag{2}$$

This target value has three error sources: the predicted future state and reward $\mathbf{s}', r \sim p_\psi(\cdot \mid \mathbf{s}, \mathbf{a})$ and future Q-value $Q_\phi(\mathbf{s}', \pi_\theta(\mathbf{s}'))$. Thus, the target value from model-generated data, $\hat{y}_{\text{model}}$, is more prone to overestimation than the original target Q-value, $\hat{y}_{\text{env}}$, computed from $(\mathbf{s}, \mathbf{a}, r, \mathbf{s}') \sim D_{\text{env}}$:

$$\hat{y}_{\text{env}} = r + \gamma Q_\phi(\mathbf{s}', \pi_\theta(\mathbf{s}')). \tag{3}$$

To mitigate the overestimation of $\hat{y}_{\text{model}}$ from inaccurate H-step model rollouts, we propose to use lower expectile regression on target Q-value estimates with small τ. As illustrated in Figure 2, expectile regression with small τ learns a Q-function predicting Q-values lower than the expectation, i.e., a *conservative estimate* of a target Q-value. Another advantage of using lower expectile regression is that we do not have to exhaustively evaluate Q-values to get τ-expectiles as Jeong et al. (2023);

instead, we can learn a conservative Q-function with sampling:

$$L_{Q,\text{model}}(\phi) = \mathbb{E}_{\mathbf{s}_0 \in \mathcal{D}_{\text{model}}, \mathcal{T} \sim p_\psi, \pi_\theta} \left[\frac{1}{H} \sum_{t=0}^{H} L_2^\tau(Q_\phi(\mathbf{s}_t, \pi_\theta(\mathbf{s}_t)) - \hat{y}_{\text{model}}) \right]. \tag{4}$$

Additionally, we train the Q-function with the transitions in the dataset $\mathcal{D}_{\text{env}}$, which do not have the risk of overestimation caused by the inaccurate model, using the standard Bellman update:

$$\mathcal{L}_{Q,\text{env}}(\phi) = \mathbb{E}_{(\mathbf{s},\mathbf{a},r,\mathbf{s}') \in \mathcal{D}_{\text{env}}} \left[\frac{1}{2} (Q_\phi(\mathbf{s}, \mathbf{a}) - \hat{y}_{\text{env}})^2 \right]. \tag{5}$$

To stabilize training of the Q-function, we adopt EMA regularization (Hafner et al., 2023), which prevents drastic change of Q-values by regularizing the difference between the predictions from Q_ϕ and ones from its exponential moving average $Q_{\bar{\phi}}$:

$$\mathcal{L}_{Q,\text{EMA}}(\phi) = \mathbb{E}_{(\mathbf{s},\mathbf{a}) \in \mathcal{D}_{\text{env}}} \left[(Q_\phi(\mathbf{s}, \mathbf{a}) - Q_{\bar{\phi}}(\mathbf{s}, \mathbf{a}))^2 \right]. \tag{6}$$

Finally, by combining the three aforementioned losses, we define the critic loss as follows:

$$\mathcal{L}_Q(\phi) = \beta \mathcal{L}_{Q,\text{model}}(\phi) + (1-\beta)\mathcal{L}_{Q,\text{env}}(\phi) + \omega_{\text{EMA}} \mathcal{L}_{Q,\text{EMA}}(\phi). \tag{7}$$

4.2 Lower expectile Q-learning with λ-return

To further improve LEQ, we use λ-return instead of 1-step return for Q-learning. λ-return allows a Q-function and policy to learn from low-bias multi-step returns (Schulman et al., 2016). Using N-step returns $G_{t:t+N}(\mathcal{T}) = \sum_{i=0}^{N-1} \gamma^i r_{t+i} + \gamma^N Q_\phi(\mathbf{s}_{t+N}, \mathbf{a}_{t+N})$, we define λ-return of an H-step trajectory $\mathcal{T}$ in timestep t, $Q_t^\lambda(\mathcal{T})$ as:[2]

$$Q_t^\lambda(\mathcal{T}) = \frac{1-\lambda}{1-\lambda^{H-t-1}} \sum_{i=1}^{H-t} \lambda^{i-1} G_{t:t+i}(\mathcal{T}). \tag{8}$$

Then, we can rewrite the Q-learning loss in Equation (4) with λ-return targets:

$$\mathcal{L}_{Q,\text{model}}^\lambda(\phi) = \mathbb{E}_{\mathbf{s}_0 \in \mathcal{D}_{\text{model}}, \mathcal{T} \sim p_\psi, \pi_\theta} \left[\sum_{t=0}^{H-1} L_2^\tau(Q_\phi(\mathbf{s}_t, \pi_\theta(\mathbf{s}_t)) - Q_t^\lambda(\mathcal{T})) \right]. \tag{9}$$

4.3 Lower expectile policy learning with λ-return

For policy optimization, we use a deterministic policy $\mathbf{a} = \pi_\theta(\mathbf{s})$ and update the policy using the deterministic policy gradients similar to DDPG (Lillicrap et al., 2016).[3] To provide more accurate learning targets for the policy, instead of maximizing the immediate Q-value, $Q_\phi(\mathbf{s}, \mathbf{a})$, we maximize the lower expectile of λ-return, analogous to our conservative critic learning in Section 4.2:

$$\mathcal{L}_\pi^\lambda(\theta) = -\mathbb{E}_{\mathbf{s}_0 \in \mathcal{D}_{\text{model}}} \left[\sum_{t=0}^{H} \mathbb{E}_{\mathcal{T} \sim p_\psi, \pi_\theta}^\tau \left[Q_t^\lambda(\mathcal{T}) \right] \right]. \tag{10}$$

However, because of the expectile term, $\mathbb{E}^\tau[Q_t^\lambda]$, we cannot directly compute the gradient of $\mathcal{L}_\pi^\lambda(\theta)$. To change the expectile to expectation, we use the relationship $\mathbb{E}^\tau[Q_t^\lambda] = \frac{\mathbb{E}\left[|\tau - \mathbb{1}(\mathbb{E}^\tau[Q_t^\lambda] > Q_t^\lambda)| \cdot Q_t^\lambda\right]}{\mathbb{E}\left[|\tau - \mathbb{1}(\mathbb{E}^\tau[Q_t^\lambda] > Q_t^\lambda)|\right]}$, and optimize the unnormalized version, $\mathbb{E}\left[|\tau - \mathbb{1}(\mathbb{E}^\tau[Q_t^\lambda] > Q_t^\lambda)| \cdot Q_t^\lambda\right]$. By approximating $\mathbb{E}^\tau[Q_t^\lambda]$ with the learned Q-estimator $Q_\phi(\mathbf{s}_t, \mathbf{a}_t)$, we derive a differentiable surrogate loss of Equation (10):

$$\hat{\mathcal{L}}_\pi^\lambda(\theta) = -\mathbb{E}_{\mathbf{s}_0 \in \mathcal{D}_{\text{model}}, \mathcal{T} \sim p_\psi, \pi_\theta} \left[\sum_{t=0}^{H} |\tau - \mathbb{1}\left(Q_\phi(\mathbf{s}_t, \mathbf{a}_t) > Q_t^\lambda(\mathcal{T})\right)| \cdot Q_t^\lambda(\mathcal{T}) \right]. \tag{11}$$

Intuitively, this surrogate loss sets a higher weight $(1-\tau)$ on a conservative λ-return estimates (i.e. $Q_\phi(\mathbf{s}_t, \mathbf{a}_t) > Q_t^\lambda(\mathcal{T})$), encouraging a policy to optimize for this conservative λ-return. On the other hand, an optimistic λ-return estimates (i.e. $Q_\phi(\mathbf{s}_t, \mathbf{a}_t) < Q_t^\lambda(\mathcal{T})$) has a less impact to the policy with a smaller weight (τ). We provide a proof in Appendix B showing that the proposed surrogate loss provides a better approximation of Equation (10) than the immediate Q-value, $Q_\phi(\mathbf{s}_t, \mathbf{a}_t)$.

[2] Our λ-return slightly differs from (Sutton, 1988) that puts a high weight to the last N-step return, $G_{t:H}(\mathcal{T})$.

[3] LEQ also works with a stochastic policy; but, a deterministic policy is sufficient for our experiments.

4.4 EXPANDING DATASET WITH MODEL ROLLOUTS

One of the problems of offline RL is that data distribution is limited to the offline dataset $\mathcal{D}_{\text{env}}$. To tackle this problem, we expand the dataset using simulated trajectories, which we refer to as $\mathcal{D}_{\text{model}}$ (Yu et al., 2021). To diversify the simulated trajectories in $\mathcal{D}_{\text{model}}$, we use a *noisy exploration policy*, which adds a noise $\epsilon \sim N(0, \sigma^2_{\text{exp}})$, to the current policy and generate a trajectory of length R.

5 EXPERIMENTS

In this paper, we propose a novel model-based offline RL method with simple, efficient, yet accurate conservative value estimation. Through our experiments, we aim to answer the following questions: (1) Can **LEQ** solve diverse domains of problems, including both dense-reward tasks and long-horizon sparse-reward tasks? (2) Can **LEQ** be applied to pixel-based environments? (3) How individual components of **LEQ** affect the performance?

5.1 TASKS

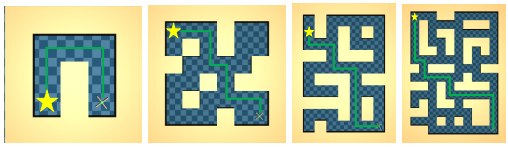
To verify the strength of our low-bias model-based conservative value estimation in diverse domains, we test **LEQ** on four benchmarks: D4RL AntMaze, D4RL MuJoCo Gym (Fu et al., 2020), NeoRL (Qin et al., 2022), and V-D4RL (Lu et al., 2023). We first test on long-horizon AntMaze tasks: `umaze`, `medium`, `large` from D4RL, and `ultra` from Jiang et al. (2023), as shown in Figure 3. We also evaluate **LEQ** on locomotion tasks (Figure 4): state-based tasks from D4RL, NeoRL and pixel-based tasks from V-D4RL. Please refer to Appendix A for more experimental details.

5.2 COMPARED OFFLINE RL ALGORITHMS

Model-free offline RL. We consider behavioral cloning (**BC**) (Pomerleau, 1989); **TD3+BC** (Fujimoto and Gu, 2021), which combines BC loss to TD3; **CQL** (Kumar et al., 2020), which penalizes out-of-distribution actions; and **IQL** (Kostrikov et al., 2022), which uses upper-expectile regression to estimate the value function. For locomotion tasks, we also compare with **EDAC** (An et al., 2021), which penalizes Q-values based on its uncertainty. For pixel-based tasks, we compare with **DrQ+BC** (Lu et al., 2023), which combines BC loss to DrQ-v2 (Yarats et al., 2022); **ACRO**, which learns representations with a multi-step inverse dynamics model.

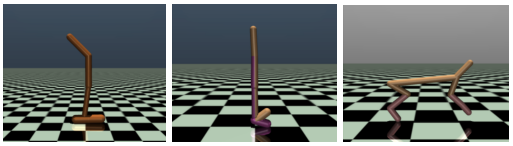
Model-based offline RL. We consider **MOPO** (Yu et al., 2020) and **MOBILE** (Sun et al., 2023), which penalize Q-values according to the transition uncertainty and the Bellman uncertainty of a world model, respectively; **COMBO** (Yu et al., 2021), which combines CQL with MBPO; **RAMBO** (Rigter et al., 2022), which trains an adversarial world model against the policy; and **CBOP** (Jeong et al., 2023), which utilizes multi-step returns for critic updates; **IQL-TD-MPC** (Chitnis et al., 2024), which extends TD-MPC (Hansen et al., 2022) to offline setting with IQL. For pixel-based environments, we consider **OfflineDV2** (Lu et al., 2023), which penalizes Q-values according to the dynamics errors, and **ROSMO** (Liu et al., 2023), which uses one-step model rollouts for policy improvement. We note that LEQ takes 2 hours with a single RTX4090 GPU for training, which is 6x faster than recent algorithms like CBOP and MOBILE. (Please refer to Table 8 for detailed comparison).

Sequence modeling for offline RL. We consider **TT** (Janner et al., 2021), which trains a Transformer model (Vaswani, 2017) to predict offline trajectories and applies beam search to find the best



(a) Umaze (b) Medium (c) Large (d) Ultra

Figure 3: AntMaze tasks.



(a) Hopper (b) Walker2d (c) HalfCheetah

Figure 4: Locomotion tasks.

Table 1: **AntMaze results.** Each number represents the average success rate on 100 trials over different seeds. The results for **LEQ**, **MOBILE**, and **CBOP** are averaged over 5 seeds. The results for other methods are reported following their respective papers.

	Model-free		Seq. modeling		Model-based					
Dataset	**CQL**	**IQL**	**TT**	**TAP**	**COMBO**	**RAMBO**	**MOBILE**†	**CBOP**†	**IQL-TD-MPC**	**LEQ (ours)**
`umaze`	74.0	87.5	**100.0**	81.5	80.3	25.0	0.0 ± 0.0	0.0 ± 0.0	52.0	94.4 ± 6.3
`umaze-diverse`	**84.0**	62.2	21.5	68.5	57.3	0.0	0.0 ± 0.0	0.0 ± 0.0	72.6	71.0 ± 12.3
`medium-play`	61.2	71.2	**93.3**	78.0	0.0	16.4	0.0 ± 0.0	0.0 ± 0.0	**88.8**	58.8 ± 33.0
`medium-diverse`	53.7	**70.0**	**100.0**	85.0	0.0	23.2	0.0 ± 0.0	0.0 ± 0.0	40.3	46.2 ± 23.2
`large-play`	15.8	39.6	66.7	**74.0**	0.0	0.0	0.0 ± 0.0	0.0 ± 0.0	66.6	58.6 ± 9.1
`large-diverse`	14.9	47.5	60.0	**82.0**	0.0	2.4	0.0 ± 0.0	0.0 ± 0.0	4.0	60.2 ± 18.3
`ultra-play`	–	8.3	20.0	22.0	–	–	0.0 ± 0.0	0.0 ± 0.0	20.6	**25.8** $\pm$ **18.2**
`ultra-diverse`	–	15.6	33.3	26.0	–	–	0.0 ± 0.0	0.0 ± 0.0	3.6	**55.8** $\pm$ **18.3**
Total w/o `ultra`	303.6	354.1	441.5	**469.0**	137.6	67.0	0.0	0.0	324.3	388.8
Total	–	378.0	**494.8**	**517.0**	–	–	0.0	0.0	348.5	470.4

† We use the official implementation of **MOBILE** and **CBOP**.

Table 2: **NeoRL results. LEQ** and **IQL** results are averaged over 5 seeds. The results for prior works are reported following Sun et al. (2023) and Qin et al. (2022). **MOPO*** is an improved version of **MOPO** presented in Sun et al. (2023). We highlight the results better than 95% of the best score.

	Model-free					Model-based		
Dataset	**BC**	**TD3+BC**	**CQL**	**EDAC**	**IQL**	**MOPO***	**MOBILE**	**LEQ (ours)**
`Hopper-L`	15.1	15.8	16.0	18.3	16.7	6.2	17.4	**24.2** ± 2.3
`Hopper-M`	51.3	70.3	64.5	44.9	28.4	1.0	51.1	**104.3** ± 5.2
`Hopper-H`	43.1	75.3	76.6	52.5	22.3	11.5	87.8	**95.5** ± 13.9
`Walker2d-L`	28.5	43.0	44.7	40.2	30.7	11.6	37.6	**65.1** ± 2.3
`Walker2d-M`	48.7	58.5	57.3	57.6	51.8	39.9	**62.2**	45.2 ± 19.4
`Walker2d-H`	**72.6**	69.6	**75.3**	**75.5**	**76.3**	18.0	**74.9**	**73.7** ± 1.1
`HalfCheetah-L`	29.1	30.0	38.2	31.3	30.7	40.1	**54.7**	33.4 ± 1.6
`HalfCheetah-M`	49.0	52.3	54.6	54.9	51.8	62.3	**77.8**	59.2 ± 3.9
`HalfCheetah-H`	71.4	75.3	77.4	**81.4**	76.3	65.9	**83.0**	71.8 ± 8.0
Total	408.8	490.1	504.6	456.6	385.0	256.5	**546.5**	**572.4**

trajectory; and **TAP** (Jiang et al., 2023), which improves TT by quantizing the action space with VQ-VAE (Van Den Oord et al., 2017).

5.3 Results on long-horizon AntMaze tasks

As shown in Table 1, **LEQ** significantly outperforms the prior *model-based* approaches. For example, **LEQ** achieves 60.2 and 55.8 for `large-diverse` and `ultra-diverse`, while the second best method, **IQL-TD-MPC** (Chitnis et al., 2024), scores only 4.0 and 3.6, respectively. We believe these performance gains come from our conservative value estimation, which works more stable than the uncertainty-based penalization of prior works. Moreover, **LEQ** even significantly outperforms the *model-free* approaches in `umaze`, `large`, and `ultra` mazes, and outperforms *sequence modeling* methods, **TT** and **TAP**, which serve as strong baselines for AntMaze tasks, in the most challenging `ultra` mazes, showing the advantage of utilizing low-bias multi-step return on long-horizon tasks.

Despite its superior performance, **LEQ** shows high variance in the performance on `antmaze-medium`. We found that the medium mazes include states separated by walls that are very close to each other (denoted as red circles in Figure 5), such that all of the learned world models falsely believe the agent can pass through the walls. This incorrect prediction makes the agent to plan faster, but impossible trajectories as shown in Figure 5. We believe that this could be addressed by employing improved world models or increasing the number of ensembles for the world models.

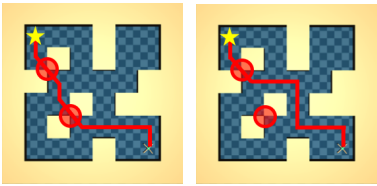


Figure 5: **Failure in medium mazes.** The agent plans impossible trajectories on certain states (red circles).

Table 3: **D4RL MuJoCo Gym results.** Each number is a normalized score averaged over 100 trials (Fu et al., 2020). Our results are averaged over 5 seeds. The results for prior works are reported following their respective papers. **MOPO*** is an improved version of **MOPO**, introduced in Sun et al. (2023). We highlight the results that are better than 95% of the best score.

	Model-free			Seq. modeling		Model-based						
Dataset	**CQL**	**EDAC**	**IQL**	**TT**	**TAP**	**MOPO***	**COMBO**	**RAMBO**	**MOBILE**	**CBOP**	**IQL-TD-MPC**	**LEQ (ours)**
`hopper-r`	5.3	25.3	7.6	6.9	-	31.7	17.9	25.4	**31.9**	**32.8**	-	**32.4** ± 0.3
`hopper-m`	61.9	**101.6**	66.3	67.4	63.4	62.8	97.2	87.0	**106.6**	**102.6**	-	**103.4** ± 0.3
`hopper-mr`	86.3	**101.0**	94.7	99.4	87.3	**99.4**	**103.5**	89.5	99.5	**104.3**	-	**103.9** ± 1.3
`hopper-me`	96.9	**110.7**	91.5	**110.0**	105.5	81.6	**111.1**	88.2	**112.6**	**111.6**	-	**109.4** ± 1.8
`walker2d-r`	5.4	16.6	5.2	5.9	-	7.4	7.0	0.0	17.9	17.8	-	**21.5** ± 0.1
`walker2d-m`	79.5	**92.5**	78.3	84.9	64.9	81.3	84.1	81.9	84.9	**87.7**	-	74.9 ± 26.9
`walker2d-mr`	76.8	87.1	73.9	89.2	66.8	85.6	56.0	89.2	89.9	92.7	-	**98.7** ± 6.0
`walker2d-me`	109.1	**114.7**	109.6	101.9	107.4	**112.9**	103.3	56.7	**115.2**	**117.2**	-	108.2 ± 1.3
`halfcheetah-r`	31.3	28.4	11.8	6.1	-	**38.5**	**38.8**	**39.5**	**39.3**	32.8	-	30.8 ± 3.3
`halfcheetah-m`	46.9	65.9	47.4	46.9	45.0	73.0	54.2	**77.9**	**74.6**	**74.3**	57.4	71.7 ± 4.4
`halfcheetah-mr`	45.3	61.3	44.2	44.1	40.8	**72.1**	55.1	**68.7**	**71.7**	66.4	49.2	65.5 ± 1.1
`halfcheetah-me`	95.0	**106.3**	86.7	95.0	91.8	90.8	90.0	95.4	**108.2**	**105.4**	44.8	**102.8** ± 0.4
Total	739.7	911.4	717.2	747.5	-	844.0	802.0	812.4	**959.5**	**953.4**	–	**923.2**

Table 4: **V-D4RL results.** We report the mean and standard deviation of returns over 3 seeds. For **ROSMO**, we replace the categorical distribution in their official code with Gaussian distribution to support continuous action spaces. For **MOPO**, we implement **MOPO** on top of DreamerV3. The results for other prior works are reported following Lu et al. (2023); Islam et al. (2023).

	Model-free				Model-based			
Method	**BC**	**CQL**	**DrQ+BC**	**ACRO**	**OfflineDV2**	**MOPO**	**ROSMO**	**LEQ**
`walker_walk-random`	2.0	14.4	5.5	0.0	**28.7**	3.2 ± 0.4	2.9 ± 0.4	22.4 ± 1.1
`walker_walk-medium`	40.9	14.8	**46.8**	**48.7**	34.1	37.1 ± 3.7	**49.8** ± 2.3	43.1 ± 3.2
`walker_walk-medium_replay`	16.5	11.4	28.7	27.8	**56.5**	11.4 ± 9.2	28.1 ± 0.9	43.0 ± 7.3
`walker_walk-medium_expert`	47.7	56.4	**86.4**	**91.4**	43.9	46.6 ± 3.5	82.9 ± 0.7	**87.2** ± 2.4
`cheetah_run-random`	0.0	5.9	5.8	0.0	**31.7**	3.2 ± 3.9	2.5 ± 0.5	14.8 ± 1.0
`cheetah_run-medium`	**51.6**	40.9	**53.0**	**52.8**	17.2	34.7 ± 6.6	**50.8** ± 2.5	37.9 ± 8.0
`cheetah_run-medium_replay`	25.0	10.7	44.8	41.7	**61.5**	30.3 ± 5.1	48.6 ± 1.5	34.3 ± 1.1
`cheetah_run-medium_expert`	**57.5**	20.9	50.6	46.6	10.4	36.6 ± 5.6	45.4 ± 2.2	25.1 ± 6.6
Total	241.2	175.4	**321.6**	**309.0**	284.0	203.3	**310.9**	**307.6**

5.4 Results on MuJoCo Gym locomotion tasks

For the NeoRL benchmark in Table 2, **LEQ** outperforms most of the prior works, especially in the Hopper and Walker2d domains. Furthermore, for D4RL MuJoCo Gym tasks in Table 3, **LEQ** achieves comparable results with the best score of prior works in 7 out of 12 tasks, while *sequence-modeling* approaches (**TT**, **TAP**) that excelled in AntMaze tasks struggle. These results show that **LEQ** serves as a general offline RL algorithm, widely applicable to various domains.

Similar to `antmaze-medium`, **LEQ** experiences high variance in MuJoCo tasks. During training, **LEQ** often achieves high performance, but then, suddenly falls back to 0, as shown in Figure 6. This is mainly because the learned models sometimes fail to capture failures (e.g. hopper and walker falling off) and predict an optimistic future (e.g. hopper and walker walking forward).

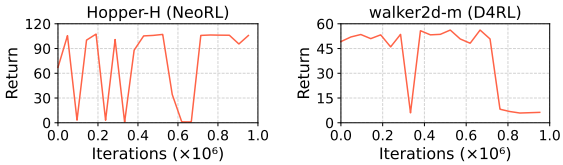


Figure 6: **High variance during training.** Our algorithm experiences oscillation due to optimistic imaginations near the initial states.

5.5 Results with visual inputs

As shown in Table 4, **LEQ** combined with DreamerV3 (Hafner et al., 2023) performs on par with the state-of-the-art methods on V-D4RL datasets, demonstrating its scalability to visual observations. Notably, **LEQ** achieves the highest score on the `walker_walk-medium_expert` dataset among model-based methods, where **OfflineDV2** and **MOPO** struggles. **LEQ** also outperforms model-free approaches on `random` datasets and `walker_walk-medium_replay` dataset, highlighting the strength of model-based methods in datasets with diverse state-action distributions.

Table 5: **Impact of lower expectile Q-learning and λ-returns on AntMaze.** We ablate the effects of lower expectile and λ-returns on the critic and policy updates in **LEQ**. The design choices from **LEQ** are colored in **blue** and other options are colored in **red**. The results are averaged over 5 seeds.

Design choices			umaze		medium		large		ultra		Total
conservatism	critic update	policy update	umaze	diverse	play	diverse	play	diverse	play	diverse	
Lower expectile	λ-returns	λ-returns	**94.4** ±6.3	**71.0** ±12.3	50.2 ±39.9	46.2 ±23.2	**58.6** ±9.1	**60.2** ±18.3	25.8 ±18.2	**55.8** ±18.3	**461.8**
Lower expectile	H-step	H-step	**93.0** ±3.4	60.7 ±10.4	46.3 ±32.4	0.0 ±0.0	**57.0** ±25.6	33.3 ±43.0	0.0 ±0.0	0.0 ±0.0	290.3
Lower expectile	1-step	$Q(\mathbf{s}, \mathbf{a})$	89.6 ±4.8	37.0 ±32.8	55.8 ±28.7	29.8 ±24.5	34.2 ±13.4	49.3 ±9.0	**42.2** ±13.2	35.6 ±13.0	373.5
Lower expectile	λ-returns	$Q(\mathbf{s}, \mathbf{a})$	81.0 ±10.5	46.2 ±16.8	**61.8** ±12.4	40.6 ±11.4	39.2 ±12.5	40.5 ±11.7	**42.8** ±21.8	47.5 ±5.9	410.3
Lower expectile	λ-returns	AWR	69.2 ±7.5	44.4 ±18.4	0.6 ±0.6	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	0.0 ±0.0	114.2
MOBILE	λ-returns	λ-returns	84.3 ±3.5	40.3 ±20.4	51.3 ±9.0	39.7 ±12.5	28.3 ±21.5	33.7 ±10.0	38.0 ±27.1	23.3 ±4.9	338.9
MOBILE	1-step	$Q(\mathbf{s}, \mathbf{a})$	59.5 ±3.5	46.5 ±1.5	57.0 ±11.0	**54.0** ±9.0	23.5 ±19.5	38.5 ±1.5	39.5 ±11.5	20.5 ±20.5	339.0

5.6 Ablation Studies

To deeply understand how **LEQ (LEQ-λ)** work, we conduct ablation studies in AntMaze environments and answer to the following five questions: (1) Is **lower expectile Q-learning** better than prior uncertainty-based penalization methods? (2) Does **lower expectile policy learning** better than existing policy learning methods? (3) Does **λ-return** help? (4) Which factor enables **LEQ** to work in AntMaze? and (5) How do **imagination length** H and **data expansion** affect the performance?

(1) Lower expectile Q-learning. We compare our *lower expectile Q-learning* with the conservative value estimator in **MOBILE** (Sun et al., 2023), which penalizes Q-values based on the standard deviation of Q-ensemble networks. In Table 5, replacing lower expectile Q-learning with **MOBILE** decreases the success rate, both with λ-returns (461.8 vs 338.9) and without them (373.5 vs 339.0).

(2) Lower expectile policy learning. We also compare our *lower expectile policy learning* with AWR (Peng et al., 2019) and directly maximizing $Q(s, a)$. As shown in Table 5, **LEQ** shows better performance compared to AWR (461.8 vs 114.2) and maximizing the Q-values (461.8 vs 410.3).

(3) λ-returns. The first three rows of Table 5 show the effect of *λ-returns* in **LEQ**. Substituting λ-returns with H-step returns (461.8 vs 290.3) or 1-step returns (461.8 vs 373.5) significantly decreases the performance. Moreover, while **LEQ** shows better performance than **MOBILE** without λ-returns (373.5 vs 339.0), the performance of **LEQ** gets significantly better with λ-returns, compared to **MOBILE** with λ-returns (461.8 vs 338.9).

(4) What makes model-based offline RL work in AntMaze? **LEQ** shows outstanding performance compared to previous offline model-based RL methods, especially in `large` and `ultra` mazes. To understand which aspects of **LEQ** enabled this success, we applied its changes to **MOBILE** and analyzed the impact. The results are detailed in Table 6.

We first re-implement **MOBILE** with some technical tricks used in **LEQ** (denoted as **MOBILE***): LayerNorm (Ba et al., 2016), SymLog (Hafner et al., 2023), single Q-network, and no target Q-value clipping. However, **MOBILE*** still achieves a barely non-zero score, 14.0.

Table 6: **Impact of hyperparameters in MOBILE* on AntMaze. MOBILE*** uses the hyperparameters from **MOBILE**: $\beta = 0.95$, $\gamma = 0.99$, and $R = 5$, whereas **LEQ** uses $\beta = 0.25$, $\gamma = 0.997$, and $R = 10$. The results show that β is the most critical hyperparameter that makes **MOBILE*** work in AntMaze.

Hyperparams.			umaze		medium		large		ultra		Total
β	γ	R	umaze	diverse	play	diverse	play	diverse	play	diverse	
0.25	0.997	10	53.8 ±26.8	**22.5** ±22.2	54.0 ±5.8	**49.5** ±6.2	**28.3** ±6.0	**28.0** ±11.4	**25.5** ±6.9	**23.8** ±15.8	**285.3**
0.25	0.997	5	**74.0** ±6.9	3.7 ±2.6	54.7 ±27.9	28.0 ±9.6	18.7 ±18.6	8.0 ±9.3	9.7 ±8.2	9.0 ±3.7	205.7
0.25	0.99	10	39.7 ±23.4	5.0 ±7.1	39.3 ±27.9	38.0 ±15.0	0.0 ±0.0	3.7 ±5.2	0.0 ±0.0	0.0 ±0.0	125.7
0.25	0.99	5	**77.0** ±6.4	20.4 ±15.7	**64.6** ±11.1	31.6 ±16.9	2.6 ±2.8	7.2 ±8.9	4.6 ±3.0	5.0 ±4.6	213.0
0.95	0.997	10	0.0 ±0.0	0.0 ±0.0	1.8 ±3.0	0.5 ±0.9	0.2 ±0.4	2.2 ±2.3	1.0 ±1.7	0.0 ±0.0	5.7
0.95	0.997	5	0.0 ±0.0	0.0 ±0.0	7.2 ±4.1	1.6 ±2.1	9.6 ±7.1	5.4 ±4.9	0.0 ±0.0	1.8 ±2.7	25.6
0.95	0.99	10	0.0 ±0.0	0.0 ±0.0	5.0 ±5.1	0.6 ±1.2	7.4 ±14.8	1.6 ±3.2	0.0 ±0.0	0.0 ±0.0	14.6
0.95	0.99	5	1.0 ±2.0	0.0 ±0.0	6.4 ±5.5	5.0 ±5.0	0.8 ±1.6	0.8 ±1.2	0.0 ±0.0	0.0 ±0.0	14.0

Table 7: **LEQ with different imagination length H and data expansion.** A longer H can mitigate critic biases, while increasing model errors, which leads to poor performance. Each number is averaged over 5 random seeds.

Dataset	$\mathbf{H = 10}$ **(ours)**	$H = 5$	$H = 15$	w/o dataset expansion
antmaze-umaze	**94.4** ±6.3	**95.2** ±1.7	**98.6** ±0.5	**97.4** ±1.4
antmaze-umaze-diverse	**71.0** ±12.3	**67.2** ±9.1	**70.7** ±15.2	63.0 ±23.2
antmaze-medium-play	58.8 ±33.0	46.4 ±31.9	**76.3** ±17.2	58.2 ±28.0
antmaze-medium-diverse	**46.2** ±23.2	18.6 ±28.7	30.3 ±40.1	28.6 ±33.7
antmaze-large-play	58.6 ±9.1	48.6 ±15.4	**62.0** ±9.9	56.0 ±9.8
antmaze-large-diverse	**60.2** ±18.3	35.2 ±8.7	33.0 ±3.2	**57.0** ±4.5
antmaze-ultra-play	25.8 ±18.2	**54.2** ±10.8	0.0 ±0.0	39.2 ±15.1
antmaze-ultra-diverse	**55.8** ±18.3	39.4 ±6.1	0.0 ±0.0	36.0 ±12.0
Total	**470.4**	404.8	371.0	435.4

We found that reducing the ratio β between the losses from imaginary rollouts and dataset transitions is key to make **MOBILE*** work (i.e. achieving meaningful performances in `umaze` and `medium` mazes, with a total score of 213.0). This adjustment also allows for a higher discount rate and longer imagination horizon, yielding the best results for **MOBILE***. We suggest that utilizing the true transition from the dataset is important in long-horizon tasks, which has been undervalued in prior works. We provide additional extensive ablation results on **LEQ** in Appendix D.

(5) Imagination length H and dataset expansion. As shown in Table 7, the performance increases when it goes to $H = 10$ from $H = 5$, but it drops when $H = 15$. This result shows the trade-off of using the world model – the further the agent imagines, the more the agent becomes robust to the error of the critic, but the more it becomes prone to the error from the model prediction.

We also evaluate **LEQ** without the dataset expansion. In AntMaze, the results with and without the dataset expansion are similar, as shown in Table 7. On the other hand, the dataset expansion makes the policy more stable and better in the D4RL MuJoCo tasks (in Appendix, Table 20).

6 Limitations

Following prior work on model-based offline RL (Sun et al., 2023; Jeong et al., 2023), we assume access to the ground-truth *termination function* of a task, different from online model-based RL approaches, which learn a termination function from interactions. As shown in Table 22, replacing the true terminal function with a learned terminal function leads to a significant performance drop (461.8 $\rightarrow$ 232.6), especially in diverse datasets (233.2 $\rightarrow$ 66.8), where termination signals are scarce because the dataset is collected by navigating to randomly selected goals. While relying on the ground-truth termination function simplifies the problem, it limits the applicability of the method to scenarios where this information is readily available. Extending the proposed approach to learn terminal signals from the dataset would be an immediate next step.

7 Conclusion

In this paper, we propose a novel offline model-based reinforcement learning method, LEQ, which uses *expectile regression* to get a *conservative evaluation* of a policy from model-generated trajectories. Expectile regression eases the pain of constructing the whole distribution of Q-targets and allows for learning a conservative Q-function via sampling. Combined with λ-returns in both critic and policy updates for the imaginary rollouts, the policy can receive learning signals that are more robust to both model errors and critic errors. We empirically show that LEQ robustly improves the performance of model-based approaches in various domains, including state-based locomotion, long-horizon navigation, and visual control tasks.

Ethics Statement

Our method aims to increase the ability of autonomous agents, such as robots and self-driving cars, to learn from static, offline data without interacting with the world. This enables autonomous agents

to utilize data with diverse qualities (not necessarily from experts). We believe that this paper does not have any immediate negative ethical concerns.

Reproducibility Statement

To ensure the reproducibility of our work, we provide the full code of LEQ in the supplementary materials, along with instructions to replicate the experiments presented in the paper. We provide the experimental details in Appendix A and the proof on the derivation of our surrogate policy objective in Appendix B.

References

Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in neural information processing systems*, 34:7436–7447, 2021.

Arthur Argenson and Gabriel Dulac-Arnold. Model-based offline planning. In *International Conference on Learning Representations*, 2021.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Philip J Ball, Laura Smith, Ilya Kostrikov, and Sergey Levine. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, pages 1577–1594. PMLR, 2023.

Rohan Chitnis, Yingchen Xu, Bobak Hashemi, Lucas Lehnert, Urun Dogan, Zheqing Zhu, and Olivier Delalleau. Iql-td-mpc: Implicit q-learning for hierarchical model predictive control. In *IEEE International Conference on Robotics and Automation*, pages 9154–9160. IEEE, 2024.

Vladimir Feinberg, Alvin Wan, Ion Stoica, Michael I Jordan, Joseph E Gonzalez, and Sergey Levine. Model-based value estimation for efficient model-free reinforcement learning. *arXiv preprint arXiv:1803.00101*, 2018.

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 *Neural Information Processing Systems*, volume 34, pages 20132–20145, 2021.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pages 2052–2062. PMLR, 2019.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870. PMLR, 2018.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2019.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control. In *International Conference on Machine Learning*, 2022.

Matteo Hessel, Joseph Modayil, Hado Van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. Rainbow: Combining improvements in deep reinforcement learning. In *Association for the Advancement of Artificial Intelligence*, 2018.

Riashat Islam, Manan Tomar, Alex Lamb, Yonathan Efroni, Hongyu Zang, Aniket Didolkar, Dipendra Misra, Xin Li, Harm Van Seijen, Remi Tachet Des Combes, et al. Principled offline rl in the presence of rich exogenous information. In *International Conference on Machine Learning*, 2023.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Neural Information Processing Systems*, volume 32, 2019.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. In *Neural Information Processing Systems*, volume 34, pages 1273–1286, 2021.

Jihwan Jeong, Xiaoyu Wang, Michael Gimelfarb, Hyunwoo Kim, Baher Abdulhai, and Scott Sanner. Conservative bayesian model-based value expansion for offline policy optimization. In *International Conference on Learning Representations*, 2023.

Zhengyao Jiang, Tianjun Zhang, Michael Janner, Yueying Li, Tim Rocktäschel, Edward Grefenstette, and Yuandong Tian. Efficient planning in a compact latent action space. In *International Conference on Learning Representations*, 2023.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. In *Neural Information Processing Systems*, volume 33, pages 21810–21823, 2020.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. In *Neural Information Processing Systems*, volume 32, 2019.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *Neural Information Processing Systems*, volume 33, pages 1179–1191, 2020.

Hoang Le, Cameron Voloshin, and Yisong Yue. Batch policy learning under constraints. In *International Conference on Machine Learning*, pages 3703–3712. PMLR, 2019.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations*, 2016.

Zichen Liu, Siyi Li, Wee Sun Lee, YAN Shuicheng, and Zhongwen Xu. Efficient offline policy optimization with a learned model. In *International Conference on Learning Representations*, 2023.

Cong Lu, Philip J Ball, Tim GJ Rudner, Jack Parker-Holder, Michael A Osborne, and Yee Whye Teh. Challenges and opportunities in offline reinforcement learning from visual observations. *Transactions on Machine Learning Research*, 2023.

Seohong Park, Dibya Ghosh, Benjamin Eysenbach, and Sergey Levine. Hiql: Offline goal-conditioned rl with latent states as actions. In *Neural Information Processing Systems*, volume 36, 2024.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems*, pages 305–313, 1989.

Rong-Jun Qin, Xingyuan Zhang, Songyi Gao, Xiong-Hui Chen, Zewen Li, Weinan Zhang, and Yang Yu. NeoRL: A near real-world benchmark for offline reinforcement learning. In *Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL `https://openreview.net/forum?id=jNdLszxdtra`.

Marc Rigter, Bruno Lacerda, and Nick Hawes. Rambo-rl: Robust adversarial model-based offline reinforcement learning. In *Neural Information Processing Systems*, volume 35, pages 16082–16097, 2022.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016.

Yihao Sun. Offlinerl-kit: An elegant pytorch offline reinforcement learning library. `https://github.com/yihaosun1124/OfflineRL-Kit`, 2023.

Yihao Sun, Jiaji Zhang, Chengxing Jia, Haoxin Lin, Junyin Ye, and Yang Yu. Model-bellman inconsistency for model-based offline reinforcement learning. In *International Conference on Machine Learning*, pages 33177–33194. PMLR, 2023.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3: 9–44, 1988.

Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

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

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

A Vaswani. Attention is all you need. *Neural Information Processing Systems*, 2017.

Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. In *International Conference on Learning Representations*, 2022.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. In *Neural Information Processing Systems*, volume 33, pages 14129–14142, 2020.

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. Combo: Conservative offline model-based policy optimization. In *Neural Information Processing Systems*, volume 34, pages 28954–28967, 2021.

A Training Details

Computing resources. All experiments are done on a single RTX 4090 GPU and 8 AMD EPYC 9354 CPU cores. For state-based environments, we use 5 different random seeds for each experiment and report the mean and standard deviation, while each offline RL experiment takes 2 hours for ours, 12 hours for MOBILE, and 24 hours for CBOP. For pixel-based environments, we use 3 different random seeds and each experiment takes 2 hours for ours and 8 hours for ROSMO.

Environment details. For state-based locomotion tasks, we use the datasets provided by D4RL (Fu et al., 2020) and NeoRL (Qin et al., 2022). Following IQL (Kostrikov et al., 2022), we normalize rewards using the maximum and minimum returns of all trajectories. We use the true termination functions of the environments, implemented in MOBILE (Sun et al., 2023). For pixel-based environments, we do not normalize the rewards.

For AntMaze tasks, we use the datasets provided by D4RL (Fu et al., 2020). Following IQL (Kostrikov et al., 2022), we subtract 1 from the rewards in the datasets so that the agent receives -1 for each step and 0 on termination. We use the true termination functions of the environments. The termination functions of the AntMaze tasks are not deterministic because a goal of a maze is randomized every time the environment is reset. Nevertheless, we follow the implementation of CBOP (Jeong et al., 2023), where the termination region is set to a circle around the mean of the goal distribution with the radius 0.5.

Implementation details of compared methods. For all compared methods, we use the results from their corresponding papers when available. For IQL (Kostrikov et al., 2022), we run the official implementation with 5 seeds to reproduce the results for the random datasets in D4RL and NeoRL. For the AntMaze tasks, we run the official implementation of MOBILE and CBOP with 5 random seeds. Please note that the original MOBILE implementation does not use the true termination function, so we replace it with our termination function. For MOPO, COMBO, and RAMBO, we use the results reported in RAMBO (Rigter et al., 2022). For DMControl tasks, we replace the categorical distribution of the policy with gaussian distribution of the official ROSMO codebase, and run the experiments with sampling hyperparameter $N = 4$.

World models. For state-based environments, we use the architecture and training script from OfflineRL-Kit (Sun, 2023), matching the implementation of MOBILE (Sun et al., 2023). Each world model is implemented as a 4-layer MLPs with the hidden layer size of 200. We construct an ensemble of world models by selecting 5 out of 7 models with the best validation scores. We pretrain the ensemble of world models for each of 5 random seeds (i.e. training in total 35 world models and using 25 models), which takes approximately 5 hours in average. For pixel-based environments, we use the 12M model of DreamerV3 (Hafner et al., 2023) and pretrain the world model using its loss function. We follow the implementation of OfflineDV2 (Lu et al., 2023), training 7 ensemble of world models for stochastic latent prediction, which takes 4 hours in average. We select one model for each step, following the design choice of MOBILE, as we found that randomly choosing a model every step can make imaginary rollouts more robust to model biases, leading to better performance.

Policy and critic networks. For state-based environments, we use 3-layer MLPs with size of 256 both for the policy network and the critic network. We use layer normalization (Ba et al., 2016) to prevent catastrophic over/underestimation (Ball et al., 2023), and squash the state inputs using symlog to keep training stable from outliers in long-horizon model rollouts (Hafner et al., 2023). For pixel-based environments, we use the architecture of the 12M model in DreamerV3 models.

Pretraining policy and critic networks. For some environments, we found that a randomly initialized policy can lead to abnormal rewards or transition prediction from the world models in the early stage, leading to unstable training. Following CBOP (Jeong et al., 2023), we pretrain a policy π_θ and a critic Q_ϕ using behavioral cloning and FQE (Le et al., 2019), respectively for state-based experiments. We use a slightly different implementation of FQE from the original implementation, where the $\arg\min$ operation is approximated with mini-batch gradient descent, similar to standard Q-learning as shown in Algorithm 2.

Algorithm 2 FQE: Fitted Q Evaluation (Le et al., 2019)

Input: Offline dataset $\mathcal{D}_{\text{env}}$, policy π_θ
1: Randomly initialize Q-function Q_ϕ
2: **while** not converged **do**
3: $\{\mathbf{s}_i, \mathbf{a}_i, r_i, \mathbf{s}'_i\}_{i=1}^N \sim \mathcal{D}_{\text{env}}$
4: $y_i = \texttt{sg}(r_i + Q_\phi(\mathbf{s}'_i, \pi_\theta(\mathbf{s}'_i)))$ ▷ $\texttt{sg}(\cdot)$ is stop-gradient operator
5: $L_{\text{FQE}}(\phi) = \frac{1}{N}\sum_{i=1}^N (Q_\phi(\mathbf{s}_i, \mathbf{a}_i) - y_i)^2$
6: Update Q_ϕ using gradient descent to minimize $L_{\text{FQE}}(\phi)$

Comparisons with prior methods. We provide a comparison of LEQ with the prior model-based approaches and the baseline methods used in our ablation studies in Table 8.

Table 8: Comparisons with the prior model-based methods and the baseline method. The hyperparameters same with LEQ are colored in **blue**; others are colored in **red**.

Components	CBOP	MOBILE	MOBILE*	LEQ (ours)
Training scheme	MVE (Feinberg et al., 2018)	MBPO (Janner et al., 2019)	MBPO (Janner et al., 2019)	Dreamer (Hafner et al., 2023)
Conservatism	Lower-confidence bound	Lower-confidence bound	Lower-confidence bound	Lower expectile
Policy	Stochastic	Stochastic	Stochastic	Deterministic
Policy objective	$Q(\mathbf{s}, \mathbf{a})$	$Q(\mathbf{s}, \mathbf{a})$	$Q(\mathbf{s}, \mathbf{a})$	λ-returns
Policy pretraining	BC	–	–	BC
# of critics	20-50	2	1	1
Critic objective	Multi-step (adaptive weighting)	One-step	One-step	λ-returns
Critic pretraining	FQE (Le et al., 2019)	–	–	FQE (Le et al., 2019)
Horizon length (H)	10	1	1	10
Expansion length (R)	–	1 or 5	10	5
Discount rate (γ)	0.99	0.99	0.997	0.997
β in Equation (7)	1.0	0.95	0.25	0.25
Impl. tricks	–	Clip Q-values with 0	LayerNorm + Symlog	LayerNorm + Symlog
Running time	24h	12h	40m	2h

Hyperparameters of LEQ. For state-based experiments, we report task-agnostic hyperparameters of our method in Table 9. We note that **we use the same hyperparameters** across all state-based tasks, except τ. We search the value of τ in $\{0.1, 0.3, 0.4, 0.5\}$ and report the best value for the main experimental results. In addition, we report the exhaustive results in Tables 15 and 16, and summarize τ used in the main results in Table 10.

For pixel-based experiments, we decrease the horizon length of DreamerV3 from 15 to 5, since we do not observe performance improvement with the longer imagination horizon (Appendix D), consistent with the finding from Lu et al. (2023). Moreover, we remove the entropy bonus, as exploration is not required in offline RL. We search the value of τ in $\{0.1, 0.3, 0.4, 0.5\}$ and report the best value in the main experimental results. All other hyperparameters follow the default settings of DreamerV3. We also report the exhaustive results in Table 17, and summarize τ in Table 13.

Table 9: Shared hyperparameters of LEQ in state-based experiments.

Hyperparameters	Value	Description
lr_{actor}	3e-5	Learning rate of actor
lr_{critic}	1e-4	Learning rate of critic
Optimizer	Adam	Optimizer
T_{expand}	5000	Interval of expanding dataset
N_{expand}	50000	Number of data for each expansion of dataset
R	5	Rollout length for dataset expansion
σ_{exp}	1.0	Exploration noise for dataset expansion
N_{iter}	1M	Total number of gradient steps.
B_{env}	256	Batch size from original dataset
B_{model}	256	Batch size from expanded dataset
γ	0.997	Discount factor
λ	0.95	λ value for λ-return
H	10	Imagination length
ω_{EMA}	1	Weight for critic EMA regularization
ϵ_{EMA}	0.995	Critic EMA decay

Table 10: Task-specific hyperparameter τ of LEQ in state-based experiments.

Domain	Task	τ
AntMaze	`umaze`	0.1
	`umaze-diverse`	0.1
	`medium-play`	0.3
	`medium-diverse`	0.1
	`large-play`	0.3
	`large-diverse`	0.3
	`ultra-play`	0.1
	`ultra-diverse`	0.1
MuJoCo	`hopper-r`	0.1
	`hopper-m`	0.1
	`hopper-mr`	0.3
	`hopper-me`	0.1
	`walker2d-r`	0.1
	`walker2d-m`	0.3
	`walker2d-mr`	0.5
	`walker2d-me`	0.1
	`halfcheetah-r`	0.3
	`halfcheetah-m`	0.3
	`halfcheetah-mr`	0.4
	`halfcheetah-me`	0.1
NeoRL	`Hopper-L`	0.1
	`Hopper-M`	0.1
	`Hopper-H`	0.1
	`Walker2d-L`	0.3
	`Walker2d-M`	0.1
	`Walker2d-H`	0.1
	`HalfCheetah-L`	0.1
	`HalfCheetah-M`	0.3
	`HalfCheetah-H`	0.3

Task-specific hyperparameters of the compared methods. We report the best hyperparameters of MOBILE* for the AntMaze tasks in Tables 11 and 12. For MOBILE and MOBILE*, we search the value of c within $\{0.1, 0.5, 1.0, 1.5\}$, as suggested in MOBILE (Sun et al., 2023), where c is the coefficient of the penalized bellman operator:

$$T\hat{Q}(\mathbf{s}, \mathbf{a}) = r(\mathbf{s}, \mathbf{a}) + \gamma Q(\mathbf{s}', \mathbf{a}') - c \cdot \text{Std}(Q(\mathbf{s}', \mathbf{a}')). \tag{12}$$

For CBOP, we conduct hyperparameter search for ψ in $\{0.5, 2.0, 3.0, 5.0\}$, as suggested in the original paper, where ψ is an LCB coefficient of CBOP. We do not report the best hyperparameter for MOBILE and CBOP because both methods score zero points for all hyperparameters in AntMaze.

For MOPO in V-D4RL experiments, we search for λ in $\{3, 10\}$, as suggested in Lu et al. (2023), where λ is the penalization coefficient in MOPO. Then, we report the best value in Table 14. For ROSMO, we use the hyperparameter specified in the official code.

Table 11: Task-specific hyperparameters in MOBILE*.

Domain	Task	c
AntMaze	umaze	1.0
	umaze-diverse	1.0
	medium-play	1.0
	medium-diverse	0.1
	large-play	0.1
	large-diverse	0.1
	ultra-play	1.0
	ultra-diverse	1.0

Table 12: Task-specific hyperparameters in MOBILE* with λ-returns.

Domain	Task	c
AntMaze	umaze	1.0
	umaze-diverse	0.5
	medium-play	0.1
	medium-diverse	0.1
	large-play	0.1
	large-diverse	0.1
	ultra-play	1.0
	ultra-diverse	0.5

Table 13: Task-specific hyperparameter τ of LEQ in V-D4RL experiments.

Domain	Task	τ
walker_walk	random	0.5
	medium	0.3
	medium_replay	0.3
	medium_expert	0.1
cheetah_run	random	0.3
	medium	0.1
	medium_replay	0.1
	medium_expert	0.5

Table 14: Task-specific hyperparameter λ of MOPO in V-D4RL experiments.

Domain	Task	λ
walker_walk	random	3.0
	medium	3.0
	medium_replay	3.0
	medium_expert	3.0
cheetah_run	random	3.0
	medium	3.0
	medium_replay	10.0
	medium_expert	10.0

B PROOF OF THE POLICY OBJECTIVE

We show that the surrogate loss in Equation (11) leads to a better approximation for the expectile of λ-returns in Equation (10) than maximizing $Q_\phi(s, a)$. In other words, we show that optimizing the following policy objective:

$$\hat{J}_\lambda(\theta) = \mathbb{E}_{\mathcal{T}\sim p_\psi,\pi_\theta}[(W^\tau(Q_\phi(\mathbf{s}_t, \mathbf{a}_t) > Q_t^\lambda(\mathcal{T}))Q_t^\lambda(\mathcal{T})], \tag{13}$$

leads to optimizing a lower-bias estimator of $\mathbb{E}^\tau_{\mathcal{T}\sim p_\psi,\pi_\theta}[Q_t^\lambda(\mathcal{T})]$ than $Q_\phi(\mathbf{s}_t, \mathbf{a}_t)$.

To show this, we first prove that $\hat{Y}_{\text{new}} = \frac{\mathbb{E}[W^\tau(Q_\phi(\mathbf{s}_t,\mathbf{a}_t)>Q_t^\lambda(\mathcal{T}))\cdot Q_t^\lambda(\mathcal{T})]}{\mathbb{E}[W^\tau(Q_\phi(\mathbf{s}_t,\mathbf{a}_t)>Q_t^\lambda(\mathcal{T}))]}$ is closer to $\mathbb{E}^\tau_{\mathcal{T}\sim p_\psi,\pi_\theta}[Q_t^\lambda(\mathcal{T})]$ than $Q_\phi(\mathbf{s}_t, \mathbf{a}_t)$. For deriving the proof, we generalize this situation to have an arbitrary distribution X and its estimate $\hat{Y}$, which corresponds to $X = Q_t^\lambda(\tau), \hat{Y} = Q_\phi(\mathbf{s}, \mathbf{a})$.

Theorem 1. *Let X be a distribution and $Y = E^\tau[X]$ be a lower expectile of X (i.e. $0 < \tau \leq 0.5$). Let $\hat{Y}$ be an arbitrary estimate of Y, and define $W^\tau(\cdot) = |\tau - \mathbb{1}(\cdot)|$. If we let $\hat{Y}_{new} = \frac{\mathbb{E}[W^\tau(\hat{Y}>X)\cdot X]}{\mathbb{E}[W^\tau(\hat{Y}>X)]}$ be a new estimate of Y, then $|Y - \hat{Y}_{new}| \leq \frac{1-2\tau}{1-\tau} \cdot |Y - \hat{Y}|$.*

Proof. Without loss of generality, we assume $\hat{Y} \geq Y$. Then, we have $\hat{Y}_{\text{new}} \geq Y$. Thus,

$$\begin{aligned}
&|\hat{Y}_{\text{new}} - Y| \\
&= \hat{Y}_{\text{new}} - Y \\
&= \frac{\mathbb{E}[W^\tau(\hat{Y} > X)\cdot X]}{\mathbb{E}[W^\tau(\hat{Y} > X)]} - \frac{\mathbb{E}[W^\tau(Y > X)\cdot X]}{\mathbb{E}[W^\tau(Y > X)]} && (\because \text{Def. of } \hat{Y}_{\text{new}} \text{ and } Y) \\
&= \frac{\mathbb{E}[W^\tau(Y > X)\cdot X] + \mathbb{E}[(1-2\tau)\cdot \mathbb{1}(Y \leq X \leq \hat{Y})X]}{\mathbb{E}[W^\tau(Y > X)] + \mathbb{E}[(1-2\tau)\cdot \mathbb{1}(Y \leq X \leq \hat{Y})]} - \frac{\mathbb{E}[W^\tau(Y > X)\cdot X]}{\mathbb{E}[W^\tau(Y > X)]} && (\because \text{Def. of } W^\tau(\cdot)) \\
&= \frac{\mathbb{E}[W^\tau(Y > X)]\mathbb{E}[(1-2\tau)\cdot \mathbb{1}(Y \leq X \leq \hat{Y})X] - \mathbb{E}[W^\tau(Y > X)X]\mathbb{E}[(1-2\tau)\cdot \mathbb{1}(Y \leq X \leq \hat{Y})]}{\mathbb{E}[W^\tau(Y > X)](\mathbb{E}[W^\tau(Y > X)] + \mathbb{E}[(1-2\tau)\cdot \mathbb{1}(Y \leq X \leq \hat{Y})])} \\
&\leq \frac{1-2\tau}{\mathbb{E}[W^\tau(Y > X)]^2} \cdot (\mathbb{E}[W^\tau(Y > X)]\mathbb{E}[\mathbb{1}(Y \leq X \leq \hat{Y})X] - \mathbb{E}[W^\tau(Y > X)X]\mathbb{E}[\mathbb{1}(Y \leq X \leq \hat{Y})] \\
&= \frac{1-2\tau}{\mathbb{E}[W^\tau(Y > X)]} \cdot (\mathbb{E}[\mathbb{1}(Y \leq X \leq \hat{Y})X] - Yp(Y \leq X \leq \hat{Y})) \\
&= \frac{(1-2\tau)p(Y \leq X \leq \hat{Y})}{\mathbb{E}[W^\tau(Y > X)]} \cdot (\mathbb{E}_{Y \leq X \leq \hat{Y}}[X] - Y) \\
&\leq \frac{(1-2\tau)p(Y \leq X \leq \hat{Y})}{\mathbb{E}[W^\tau(Y > X)]} \cdot (\hat{Y} - Y) \\
&= \frac{(1-2\tau)p(Y \leq X \leq \hat{Y})}{\tau + (1-2\tau)p(Y \leq X)} \cdot (\hat{Y} - Y) \\
&\leq \frac{(1-2\tau)p(Y \leq X)}{\tau + (1-2\tau)p(Y \leq X)} \cdot (\hat{Y} - Y) \\
&\leq \frac{1-2\tau}{1-\tau} \cdot |\hat{Y} - Y| && (\text{Equality holds when } p(Y \leq X) = 1)
\end{aligned}$$

□

Note that this theorem shows that the bias of the new estimate is always smaller than the original estimate, since $\frac{1-2\tau}{1-\tau} < 1$. If we plug in the distribution of $Q_t^\lambda(\mathcal{T})$ to X and $\hat{Y} = Q_\phi(s_t, \mathbf{a}_t)$, then $Y = \mathbb{E}^\tau[X] = \mathbb{E}^\tau_{\mathcal{T}\sim p_\psi,\pi_\theta}[Q_t^\lambda(\mathcal{T})]$, and we can show the desired result using the theorem: $\hat{Y}_{\text{new}} = \frac{\mathbb{E}[W^\tau(Q_\phi(\mathbf{s}_t,\mathbf{a}_t)>Q_t^\lambda(\mathcal{T}))\cdot Q_t^\lambda(\mathcal{T})]}{\mathbb{E}[W^\tau(Q_\phi(\mathbf{s}_t,\mathbf{a}_t)>Q_t^\lambda(\mathcal{T}))]}$ is closer to $\mathbb{E}^\tau_{\mathcal{T}\sim p_\psi,\pi_\theta}[Q_t^\lambda(\mathcal{T})]$ than $Q_\phi(\mathbf{s}_t, \mathbf{a}_t)$.

Here, the normalizing factor $\mathbb{E}[W^\tau(Q_\phi(\mathbf{s}_t, \mathbf{a}_t) > Q_t^\lambda(\mathcal{T}))]$ is non-differentiable with $\mathcal{T}$. Specifically, the gradient is 0 everywhere (except $Q_\phi(\mathbf{s}_t, \mathbf{a}_t) = Q_t^\lambda(\mathcal{T})$). Thus, if we calculate the gradient of $\hat{Y}_{\text{new}}$, the gradient for the normalizing factor disappears. Therefore, we can omit the normalizing factor and get an equivalent formula $\mathbb{E}[W^\tau(Q_\phi(\mathbf{s}_t, \mathbf{a}_t) > Q_t^\lambda(\mathcal{T})) \cdot Q_t^\lambda(\mathcal{T})]$ for gradient-based optimization.

C RESULTS FOR ALL EXPECTILES τ

To give insights how the expectile parameter τ affects the performance of LEQ, we report the performance of LEQ with all expectile values $\{0.1, 0.3, 0.4, 0.5\}$. The expectile parameter τ has a trade-off – high expectile makes the model's predictions less conservative while making a policy easily exploit the model. We recommend first trying $\tau = 0.1$, which works well for most of the tasks, and increase τ until the performance starts to drop.

Table 15: **Antmaze results of LEQ with different expectiles.** We report the results in Antmaze task with expectiles value of 0.1, 0.3, 0.4, 0.5. The best value is highlighted.

Expectile	0.1	0.3	0.4	0.5
antmaze-umaze	**94.4** ±6.3	39.0 ±28.1	0.2 ±0.4	3.0 ±5.5
antmaze-umaze-diverse	**71.0** ±12.2	23.6 ±21.7	4.0 ±4.2	0.0 ±0.0
antmaze-medium-play	50.2 ±39.9	**58.8** ±33.0	36.0 ±21.8	0.6 ±1.2
antmaze-medium-diverse	**46.2** ±23.2	13.2 ±13.3	11.6 ±14.8	10.6 ±13.3
antmaze-large-play	42.0 ±30.6	**58.6** ±9.1	52.2 ±15.8	42.2 ±7.3
antmaze-large-diverse	**60.6** ±32.1	60.2 ±18.3	48.8 ±5.8	36.8 ±9.7
antmaze-ultra-play	**25.8** ±18.2	10.8±8.8	11.6 ±12.5	9.2 ±11.5
antmaze-ultra-diverse	**55.8** ±18.3	4.6±3.4	7.6 ±7.3	0.6 ±1.2

Table 16: **D4RL mujoco results of LEQ with different expectiles.** We report the results in D4RL mujoco task with expectiles value of 0.1, 0.3, 0.4, 0.5. The best value is highlighted.

Expectile	0.1	0.3	0.4	0.5
hopper-r	**32.4** ±0.3	13.7 ±9.1	16.4 ±9.3	12.5 ±10.1
hopper-m	**103.4** ±0.3	102.7 ±1.7	81.4 ±24.8	38.6 ±29.2
hopper-mr	103.2 ±1.0	**103.9** ±1.3	71.5 ±34.7	103.8 ±1.9
hopper-me	**109.4** ±1.8	108.0 ±8.7	64.2 ±35.8	33.7 ±0.5
walker2d-r	**21.5** ±0.1	21.5 ±0.5	14.0 ±8.8	8.7 ±6.7
walker2d-m	26.3 ±37.4	**74.9** ±26.9	60.3 ±40.9	34.8 ±34.3
walker2d-mr	48.6 ±19.5	60.5 ±27.4	88.5 ±3.5	**98.7** ±6.0
walker2d-me	**108.2** ±1.3	98.8 ±28.8	105.8 ±25.9	33.7 ±31.9
halfcheetah-r	23.8 ±1.8	**30.8** ±3.3	29.0 ±2.9	30.2 ±2.5
halfcheetah-m	65.3 ±2.0	**71.7** ±4.4	58.5 ±23.8	55.5 ±16.7
halfcheetah-mr	60.6 ±1.4	55.4 ±27.3	**65.5** ±1.1	52.4 ±26.7
halfcheetah-me	**102.8** ±0.4	81.5 ±19.6	58.1 ±26.1	46.3 ±17.7

Table 17: **V-D4RL results of LEQ with different expectiles.** We report the results in Antmaze task with expectiles value of 0.1, 0.3, 0.4, 0.5. The best value is highlighted.

Expectile	0.1	0.3	0.4	0.5
walker_walk-random	14.5 ±1.1	20.2 ±3.6	**22.4** ±1.1	21.8 ±0.8
walker_walk-medium	**43.1** ±3.2	37.2 ±5.3	35.0 ±3.7	26.6 ±1.9
walker_walk-medium_replay	40.6 ±6.6	40.8 ±3.4	41.3 ±6.3	**43.0** ±7.3
walker_walk-medium_expert	**87.2** ±2.4	82.7 ±5.0	77.0 ±9.2	85.0 ±1.6
cheetah_run-random	12.1 ±1.9	**14.8** ±1.0	14.2 ±1.6	14.6 ±2.9
cheetah_run-medium	25.0 ±5.9	**37.9** ±8.0	23.6 ±11.9	32.2 ±6.2
cheetah_run-medium_replay	**34.3** ±1.1	32.3 ±2.1	31.2 ±3.5	31.2 ±0.4
cheetah_run-medium_expert	23.9 ±4.7	18.9 ±3.9	**25.1** ±6.6	24.3 ±7.6
Total	280.7	284.8	269.9	278.6

D MORE ABLATION RESULTS

Ablation of each component in LEQ. In Table 18, we ablate LEQ's design choices on AntMaze tasks to investigate their importance. Notably, the performance of LEQ drops significantly if we remove the Q-learning loss (461.8 to 312.8) or apply expectile regression for real transitions (461.8 to 249.2), showing the importance of utilizing the real transitions in long-horizon tasks. Moreover, the expectile loss from the policy learning is crucial (461.8 to 357.0), since simply maximizing λ-returns, even if $Q(\mathbf{s}, \mathbf{a})$ is conservative, can lead to a suboptimal policy due to the lack of conservatism in the rewards from model rollouts. Minor design choices, such as EMA regularization and dataset expansion, have a small impact on the performance in AntMaze.

Table 18: **Ablation studies about various components in LEQ on AntMaze.** We conduct ablation studies on (1) EMA regularization, (2) dataset expansion, (3) utilizing data transitions, (4) lower expectile policy optimization, (5) lower expectile regression for data transitions

Method	umaze		medium		large		ultra		Total
	umaze	diverse	play	diverse	play	diverse	play	diverse	
LEQ	**94.4** ±6.3	71.0 ±12.3	50.2 ±39.9	**46.2** ±23.2	58.6 ±9.1	60.2 ±18.3	25.8 ±18.2	**55.8** ±18.3	**461.8**
- EMA regularization	**96.0** ±4.0	65.6 ±5.1	33.8 ±31.4	21.2 ±26.1	58.8 ±8.2	62.2 ±15.0	8.0 ±13.9	33.5 ±35.1	379.1
- dataset expansion	**97.4** ±1.4	63.0 ±23.2	**58.2** ±28.0	28.6 ±33.7	56.0 ±9.8	57.0 ±4.5	**39.2** ±15.1	36.0 ±12.0	435.4
- data transitions	63.0 ±19.8	**75.3** ±4.2	31.8 ±34.0	28.0 ±27.0	50.5 ±20.9	57.5 ±8.6	0.8 ±1.3	6.0 ±5.1	312.8
- expectile in policy update	93.4 ±4.2	43.4 ±26.1	45.8 ±21.8	11.4 ±18.6	53.8 ±7.5	54.0 ±6.9	33.8 ±11.6	21.4 ±13.2	357.0
+ expectile in data transitions	53.4 ±18.8	4.8 ±7.7	27.2 ±37.0	19.2 ±11.9	**69.0** ±17.4	**75.4** ±6.4	0.0 ±0.0	0.0 ±0.0	249.2

Ablation on pretraining. Table 19 shows the effect of BC and FQE pretraining for LEQ and MOBILE* in AntMaze tasks. Without pretraining, the performance of LEQ decreases (461.8 $\rightarrow$ 322.2), especially in medium mazes (96.4 $\rightarrow$ 4.6) and ultra mazes (81.6 $\rightarrow$ 12.4) due to early instability of training, matching the observation in CBOP (Jeong et al., 2023). However, for MOBILE*, pretraining worsens the performance (285.3 $\rightarrow$ 232.2).

Table 19: **Ablation results for pretraining on AntMaze.** Results are averaged over 5 random seeds.

Method		umaze		medium		large		ultra		Total
conservatism	pretrain	umaze	diverse	play	diverse	play	diverse	play	diverse	
LEQ	O	**94.4** ±6.3	71.0 ±12.3	50.2 ±39.9	**46.2** ±23.2	58.6 ±9.1	60.2 ±18.3	25.8 ±18.2	**55.8** ±18.3	**461.8**
LEQ	X	**94.0** ±1.9	65.6 ±6.8	0.8 ±1.6	3.8 ±7.6	60.4 ±13.4	59.8 ±12.5	11.6 ±12.4	26.2 ±22.4	322.2
MOBILE*	O	73.4 ±12.6	46.0 ±9.4	24.6 ±23.2	11.6 ±9.9	31.0 ±8.4	33.2 ±10.6	12.4 ±6.9	0.0 ±0.0	232.2
MOBILE*	X	53.8 ±26.8	22.5 ±22.2	54.0 ±5.8	**49.5** ±6.2	28.3 ±6.0	28.0 ±11.4	**25.5** ±6.9	23.8 ±15.8	**285.3**

Ablation study on dataset expansion. Table 20 shows the ablation results on the dataset expansion in D4RL MuJoCo tasks. The results show that the dataset expansion generally improves the performance, especially in Hopper environments.

Table 20: **D4RL MuJoCo ablation results for dataset expansion.** Results are averaged over 5 random seeds. The dataset expansion generally improves the performance of LEQ.

Dataset	LEQ (ours)	LEQ w/o Dataset Expansion
hopper-r	**32.4** ±0.3	17.6 ±8.6
hopper-m	**103.4** ±0.3	52.7 ±45.3
hopper-mr	**103.9** ±1.3	**103.7** ±1.3
hopper-me	**109.4** ±1.8	79.7 ±42.4
walker2d-r	**21.5** ±0.1	**20.5** ±2.2
walker2d-m	74.9 ±26.9	**87.2** ±4.3
walker2d-mr	**98.7** ±6.0	78.7 ±35.5
walker2d-me	**108.2** ±1.3	**110.4** ±0.8
halfcheetah-r	**30.8** ±3.3	**27.7** ±2.2
halfcheetah-m	**71.7** ±4.4	**71.6** ±3.8
halfcheetah-mr	**65.5** ±1.1	54.4 ±26.3
halfcheetah-me	**102.8** ±0.4	83.9 ±28.0
Total	**923.2**	788.2

Ablation on using deterministic policy. We parameterize the stochastic policy $\pi(\cdot|s)$ as $\pi(a|s) = \tanh(N(\mu(s), \sigma^2(s)))$ as Haarnoja et al. (2018) and run LEQ with this configuration. However, we found that the policy quickly becomes deterministic. This is because LEQ penalizes the uncertainty of the model rollouts, inadvertently penalizing also the stochasticity of the policy. To compensate for this effect, we use an entropy bonus coefficient $\alpha = 0.0003$. As demonstrated in Table 21, stochastic policy shows slightly worse performance compared to deterministic policy (461.8 $\rightarrow$ 380.4). However, we believe that LEQ can be extended to stochastic policies with further hyperparameter tuning on the stochasticity of the policy.

Table 21: **Ablation results on using deterministic policy.** Results are averaged over 5 random seeds.

Policy	umaze		medium		large		ultra		Total
	umaze	diverse	play	diverse	play	diverse	play	diverse	
Deterministic	**94.4** ±6.3	**71.0** ±12.3	**50.2** ±39.9	**46.2** ±23.2	**58.6** ±9.1	**60.2** ±18.3	25.8 ±18.2	**55.8** ±18.3	**461.8**
Stochastic	**91.0** ±7.9	**70.4** ±6.7	35.8 ±29.7	0.0 ±0.0	41.8 ±16.5	50.2 ±7.8	**43.4** ±25.0	47.8 ±23.5	380.4

Ablation on using learned terminal function. We conduct an ablation study to evaluate the impact of using learned terminal functions instead of the ground-truth terminal function. For the terminal prediction network, we use a 3-layer MLP with a hidden size of 256, consistent with the architecture of the policy and critic networks. As shown in Table 21, replacing the true terminal function with a learned terminal function leads to a significant drop in performance.

Table 22: **Ablation results on using learned terminal function.** Results are averaged over 5 random seeds.

Terminal	umaze		medium		large		ultra		Total
	umaze	diverse	play	diverse	play	diverse	play	diverse	
Groundtruth	**94.4** ±6.3	**71.0** ±12.3	50.2 ±39.9	**46.2** ±23.2	**58.6** ±9.1	**60.2** ±18.3	**25.8** ±18.2	**55.8** ±18.3	**461.8**
Learned	66.8 ±5.2	44.0 ±22.3	**54.0** ±31.4	2.0 ±4.0	29.4 ±8.1	0.0 ±0.0	15.6 ±5.9	20.8 ±5.2	232.6

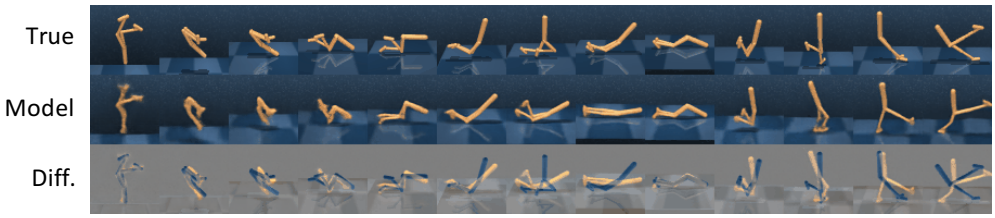
Ablation study on horizon length in V-D4RL. Table 23 shows that when we use the default hyperparameter of DreamerV3, $H = 15$, the performance drops in V-D4RL. The result suggests that we need to use a shorter imagination horizon for offline model-based RL.

Table 23: **V-D4RL ablation results for horizon length.** Results are averaged over 3 random seeds. $H = 5$ generally improves the performance of LEQ.

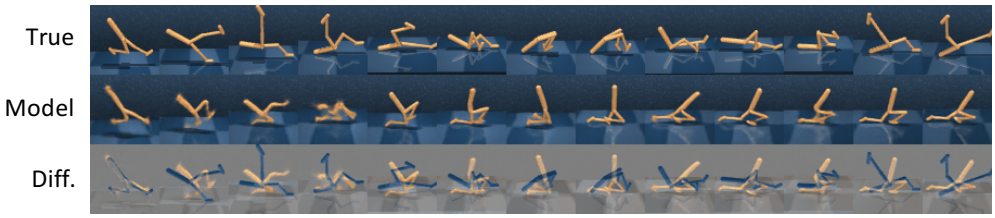
Dataset	H=5	H=15
walker_walk-random	**23.2** ±1.1	15.4 ±1.5
walker_walk-medium	**50.0** ±3.6	40.0 ±3.6
walker_walk-medium_replay	**44.0** ±28.5	**45.4** ±7.5
walker_walk-medium_expert	**90.3** ±1.8	79.9 ±1.4
cheetah_run-random	**15.3** ±1.7	**15.5** ±3.0
cheetah_run-medium	**40.1** ±14.6	32.2 ±4.3
cheetah_run-medium_replay	**39.9** ±2.0	36.4 ±2.1
cheetah_run-medium_expert	27.7 ±12.6	**29.3** ±4.1
Total	**330.5**	294.1

E VIDEO PREDICTIONS ON V-D4RL

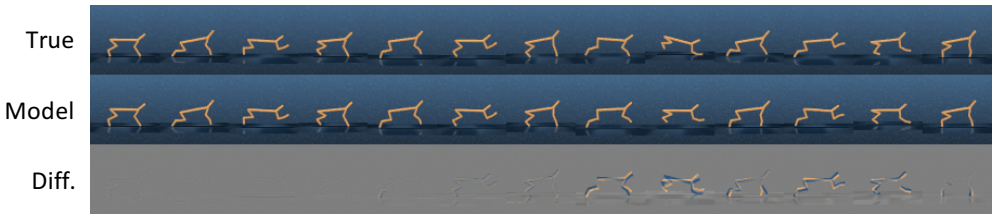
Figure 7 presents the imagined trajectory generated by the action sequence during evaluation in the V-D4RL datasets: `walker_walk-medium_replay`, `walker_walk-medium_expert`, `cheetah_run-medium_replay`, `cheetah_run-medium_expert`, from the top to the bottom. Overall, world models trained in `medium_replay` datasets show better prediction compared to `medium_expert` dataset, likely due to their broader state distribution of the dataset. Nevertheless, LEQ achieves high performance on the `walker_walk-medium_expert` dataset, despite inaccurate predictions, highlighting robustness of LEQ in handling imperfect world models.



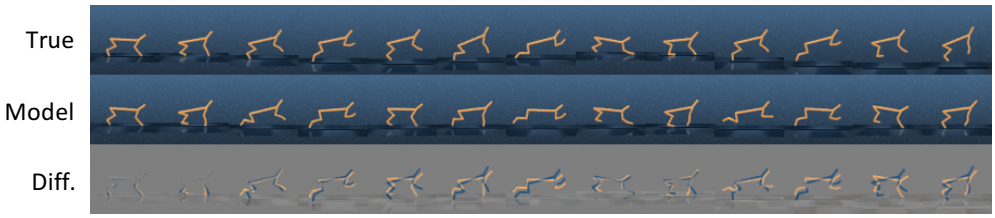
(a) `walker_walk-medium_replay`



(b) `walker_walk-medium_expert`



(c) `cheetah_run-medium_replay`



(d) `cheetah_run-medium_expert`

Figure 7: **Video prediction results in V-D4RL.** The world model receives initial 5 frames and simulates 64 additional frames based on the action sequence during evaluation. For each trajectory, each row displays the true image, model prediction, and difference between these two, respectively.