Offline Reinforcement Learning with Universal Horizon Models

Hojun Chung [*1] **Junseo Lee** [*1] **Songhwai Oh** [1 2]

Abstract

Model-based reinforcement learning (RL) offers a compelling approach to offline RL by enabling value learning on imagined on-policy trajectories. However, it often suffers from compounding errors due to repeated model inference on self-generated states. While geometric horizon models (GHM) alleviate this issue through direct prediction over a discounted infinite-horizon future, they remain challenged in accurately modeling distant future states. To this end, we introduce universal horizon models (UHM), a generalization of GHM that directly predicts future states under arbitrary horizons. Leveraging this flexibility, we propose a scalable value learning method that employs a winsorized horizon distribution to stabilize training by capping excessively large horizons. Experimental results on 100 challenging OGBench tasks demonstrate that the proposed method outperforms competitive baselines, particularly on tasks with highly suboptimal datasets and those requiring long-horizon reasoning. Project page: https://rllab-snu.github.io/projects/UHM/

1. Introduction

Offline reinforcement learning (Levine et al., 2020; Prudencio et al., 2023) provides a promising way to learn effective policies without online exploration, enabling the utilization of pre-collected datasets. However, its scalability is often restricted in tasks that require long-horizon reasoning due to the bias accumulation in temporal difference (TD) learning (Rosete-Beas et al., 2023; Park et al., 2025b). Recent works reveal the potential to solve such complex tasks using n-step TD, which utilizes n-step returns for Bellman backups (Sutton et al., 1998) to reduce the bias in the TD target (Park et al., 2025b; 2026b). Despite the benefit, model-free approaches should rely on trajectories in the dataset since they have no ability to imagine on-policy rollouts. This distributional mismatch ruins a theoretical foundation of TD learning, and can lead to inaccurate value estimation (De Asis et al., 2018; Hernandez-Garcia & Sutton, 2019).

Model-based reinforcement learning (MBRL) can address this issue with model-based value expansion, i.e., on-policy value learning on synthetic trajectories (Feinberg et al., 2018; Hafner et al., 2025). However, generating such trajectories with single-step dynamics models requires repeated model inference on self-generated states, where keeping the dynamics error small is challenging. While geometric horizon models (GHM) mitigate this issue by directly predicting future states, they lack the ability to predict future states at a specified timestep. This limitation forces GHM to model long-horizon tails of the geometric distribution, which is inherently difficult to learn accurately.

In this work, we focus on developing a scalable model-based value learning method for offline RL. We first propose universal horizon models (UHM), which can sample states directly from n-step future state distributions for any given n. As illustrated in Figure 1, UHM generalizes both GHM and single-step models, as it allows n to be sampled from arbitrary horizon distributions. This generalization enables a TD learning method that allows flexible control over which future horizons to focus on. Building on this framework, we present a value learning method using winsorized future horizons to stabilize the learning process by capping the maximum value of n. Through extensive experiments, we demonstrate that the proposed method outperforms baselines across 100 challenging tasks in OGBench (Park et al., 2025a), including tasks that provide highly suboptimal datasets or require long-horizon reasoning. In summary, the main contributions of the paper are as follows:

- We propose universal horizon models (UHM), which generalize geometric horizon models by directly sampling n-step future states for any given n.
- We introduce a robust and scalable model-based value expansion method using UHM.
- We demonstrate that the proposed method achieves a 14% higher average success rate than the strongest baseline across 100 challenging OGBench tasks.

[*]Equal contribution [1]Interdisciplinary Program in Artificial Intelligence and ASRI, Seoul National University [2]Department of Electrical and Computer Engineering, Seoul National University. Correspondence to: Songhwai Oh <songhwai@snu.ac.kr>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

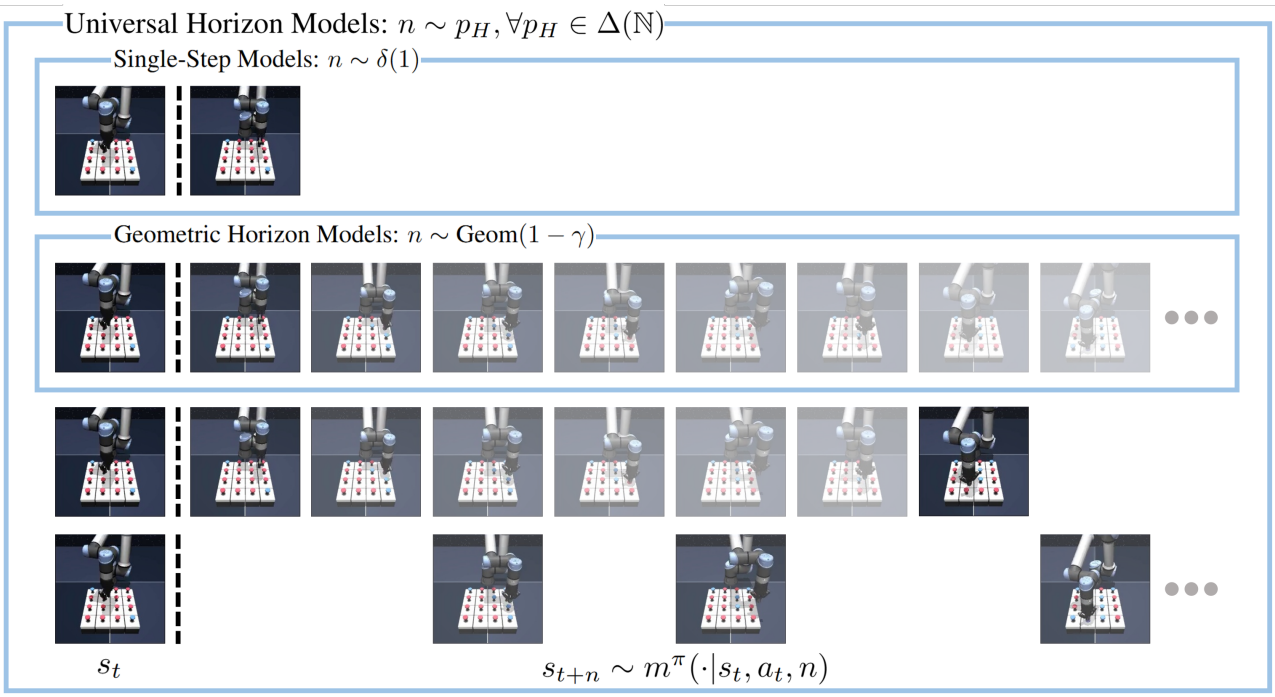

Figure 1. Universal horizon model is a future predictive model that directly samples states from the n-step future state distribution of the policy for any given horizon n. Since it allows n to be sampled from arbitrary distributions, UHM can be seen as a general framework that includes single-step models and geometric horizon models.

2. Related Work

Offline RL (Levine et al., 2020) studies the problem of learning policies from static datasets without online interaction. A wide range of offline RL algorithms are built upon temporal difference (TD) learning, which updates the critic using one-step lookahead value estimation. To adapt TD learning to offline settings, prior work has proposed uncertainty-aware methods (An et al., 2021; Ghasemipour et al., 2022; Wu et al., 2021), conservative updates (Kumar et al., 2020; Sikchi et al., 2024), and in-sample maximization techniques (Kostrikov et al., 2022; Garg et al., 2023; Xu et al., 2023). Policies are then learned from the critic via advantage-weighted regression (Peters & Schaal, 2007; Peng et al., 2019), behavior-regularized policy gradients (Fujimoto & Gu, 2021; Tarasov et al., 2023; Park et al., 2025c), or using generative modeling (Chen et al., 2023; Hansen-Estruch et al., 2023). Recently, horizon reduction techniques have demonstrated strong performance on complex tasks by training hierarchical policies (Rosete-Beas et al., 2023; Singh et al., 2021), leveraging action chunking (Seo & Abbeel, 2025; Li et al., 2025; 2026), or adopting multi-step TD methods (Park et al., 2025b; 2026b).

Offline MBRL has been studied with its potential to generate on-policy trajectories without environment interaction. Despite its success in online settings (Janner et al., 2019; Hansen et al., 2024; Hafner et al., 2025), dynamics models learned from static datasets have inevitable errors (Talvitie, 2014), making it challenging to directly apply it to offline settings. To resolve this issue, previous works consider uncertainty penalization (Kidambi et al., 2020; Yu et al., 2020; Sun et al., 2023), or conservative critic learning (Park & Lee, 2025). Another branch of MBRL leverages generative models to learn a distribution of trajectories (Janner et al., 2021; 2022; Hong et al., 2023), and guide them to produce trajectories with higher expected returns (Ajay et al., 2023; Jackson et al., 2024; Cheng et al., 2025). Meanwhile, geometric horizon models (Janner et al., 2020; Thakoor et al., 2022) have emerged as an approach for directly predicting states from the discounted future. Combined with flow-matching (Lipman et al., 2023), it shows superior performance on both future prediction and value estimation (Farebrother et al., 2025; Zheng et al., 2026). Our work is also closely related to the prior works that employ dynamics models to predict future beyond the next timestep (Zhang et al., 2023; Lin et al., 2025; Park et al., 2026a). These methods train models to predict $\Pr(s_{t+n}|s_t, a_t, \ldots, a_{t+n-1})$, which is a future determined by fixed-length action chunks. In contrast, our approach learns $\Pr(s_{t+n}|s_t, a_t, \pi)$ for an arbitrary horizon n, which is a future state distribution induced by the policy. This distinction allows us to generate horizon-flexible future states without repeated inference, thereby reducing computational overhead and avoiding recursive conditioning on self-generated states.

3. Preliminaries

Reinforcement learning (RL) can be formulated as an infinite horizon Markov decision process $(\mathcal{S}, \mathcal{A}, R, \gamma, \mathcal{P}, \rho)$, where $\mathcal{S}$ is a state space, $\mathcal{A}$ is an action space, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is a reward function, $\gamma \in (0, 1)$ is a discount factor, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is a transition probability, and $\rho \in \Delta(\mathcal{S})$ is an initial state distribution. Offline RL aims to find a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ that maximizes a cumulative discounted return $\mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R(s_k, a_k) \mid s_0 \sim \rho(\cdot), a_k \sim \pi(\cdot|s_k)\right]$ from the offline dataset $\mathcal{D} = \left\{(s_k^m, a_k^m, s_{k+1}^m)_{k=0}^{H_m}\right\}_{m \in \{1,2,\dots M\}}$.

3.1. Temporal Difference Learning for Offline RL

Value learning in offline RL is typically based on temporal difference (TD) learning. Given a transition tuple $(s, a, s') \sim \mathcal{D}$, a critic $Q(s, a)$ is trained to approximate $Q^\pi(s, a) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k r_k | s_0 = s, a_0 = a, \pi\right]$ by regressing towards its bootstrapping target $R(s, a) + \gamma Q(s', a')$, where $a' \sim \pi(\cdot|s')$. Besides its convergence guarantee, TD learning often suffers from bias accumulation, which originates from inaccurate value estimates in the bootstrapping target (Sutton et al., 1998).

A common approach to reduce the bias is n-step TD, which calculates the bootstrapping target based on the trajectory $(s_k, a_k, s_{k+1})_{k=0}^{n-1}$ whose length is $n \in \mathbb{N}$:

$$G^{(n)} = \sum_{k=0}^{n-1} \gamma^k R(s_k, a_k) + \gamma^n Q(s_n, a_n), \tag{1}$$

where $a_n \sim \pi(\cdot|s_n)$. TD(λ) generalizes n-step TD by using a weighted average of bootstrapping targets from various $n \in \mathbb{N}$ with decay parameter $\lambda \in [0, 1]$:

$$G^\lambda = (1 - \lambda) \sum_{n=1}^{\infty} \lambda^{n-1} G^{(n)}. \tag{2}$$

Both n-step TD and TD(λ) are common techniques to boost the performance in online RL (Schulman et al., 2015; Hessel et al., 2018), and recent works have shown that using n-step returns is also helpful to solve complex, long-horizon reasoning tasks in offline settings (Park et al., 2025b; 2026b). However, model-free approaches rely on behavior trajectories in offline settings, making it unclear which policy the learned value function estimates (De Asis et al., 2018; Hernandez-Garcia & Sutton, 2019).

Model-based RL can resolve this issue by performing model-based value expansion (MVE) (Feinberg et al., 2018), i.e., TD learning on synthetic on-policy rollouts. In contrast to the success in online RL (Hafner et al., 2025), dynamics models learned from static datasets are not globally accurate, exacerbating the compounding error problem induced by repeated inference of the dynamics model and policy. As a result, offline MBRL methods are often limited to restricted forms, such as conservative critic updates or short-horizon imagination (Kidambi et al., 2020; Yu et al., 2020; Sun et al., 2023; Park & Lee, 2025).

3.2. Geometric Horizon Models

Geometric horizon model (GHM) (Janner et al., 2020) is defined as a generative model of normalized successor measures (Dayan, 1993; Blier et al., 2021):

$$m^\pi(x|s, a) = (1 - \tilde{\gamma}) \sum_{k=0}^{\infty} \gamma^k \Pr(s_{k+1} = x \mid s_0 = s, a_0 = a, \pi), \tag{3}$$

which is a $\tilde{\gamma}$-discounted distribution of future states when deploying the policy π, where $\tilde{\gamma} \in (0, \gamma]$. The GHM can be learned from off-policy transitions via regression towards its bootstrapping target:

$$\begin{aligned} m^\pi(x|s, a) \leftarrow (1 - \tilde{\gamma})\mathcal{P}(x|s, a) \\ + \tilde{\gamma}\mathbb{E}_{\substack{s' \sim \mathcal{P}(\cdot|s,a) \\ a' \sim \pi(\cdot|s')}} \left[m^\pi(x|s', a')\right], \end{aligned} \tag{4}$$

which is a contraction mapping and applicable for various generative modeling methods (Thakoor et al., 2022; Farebrother et al., 2025).

Previous works propose γ-MVE (Janner et al., 2020; Thakoor et al., 2022) to obtain bootstrapping value target with future states sampled by GHM. The corresponding value target can be simplified as follows:

$$\begin{aligned} Q_{\gamma\text{-MVE}} = R(s, a) + \gamma \mathop{\mathbb{E}}_{\substack{s_e \sim m^\pi(\cdot|s,a) \\ a_e \sim \pi(\cdot|s_e)}} \Bigg[\frac{1}{1 - \tilde{\gamma}} R(s_e, a_e) \\ + \frac{\gamma - \tilde{\gamma}}{1 - \tilde{\gamma}} Q(s_e, a_e)\Bigg]. \end{aligned} \tag{5}$$

We found that $Q_{\gamma\text{-MVE}}$ (5) is equal to the expectation of TD(λ) target (2) on on-policy trajectories when $\lambda = \tilde{\gamma}/\gamma$, which means γ-MVE is a method to conduct TD(λ) using GHM. Detailed derivation is provided in Appendix A.1.

4. Proposed Methods

In this section, we introduce a *universal horizon model* (UHM), a future predictive model that generalizes geometric horizon models and single-step dynamics models. UHM directly samples n-step future states, and allows the horizon n to be sampled from arbitrary distributions. This flexibility enables a general form of value estimation that recovers both n-step TD and TD(λ). Based on this framework, we propose a scalable offline RL algorithm that stabilizes the learning process by capping excessively large horizons.

4.1. Universal Horizon Models

While GHMs avoid repeated inference on self-generated intermediate states by directly sampling from the successor measure (3), they do not reveal how many steps into the future a sampled state lies. Moreover, their horizons are restricted to a single geometric distribution. As a result, they require accurate modeling of the long-horizon tail, which is inherently difficult to learn. To generalize beyond the implicit geometric horizon, we define a universal horizon model as a generative model of n-step transition measure:

$$m^\pi(x \mid s, a, n) = \Pr(s_n = x \mid s_0 = s, a_0 = a, \pi), \quad (6)$$

where the horizon n can be sampled from any distribution p_H. The resulting marginal measure

$$m^{\pi, p_H}(x \mid s, a) = \sum_{k \geq 1} p_H(k) m^\pi(x \mid s, a, k), \quad (7)$$

$$= \sum_{k \geq 1} p_H(k) \Pr(s_k = x \mid s_0 = s, a_0 = a, \pi), \quad (8)$$

represents a horizon-weighted visitation measure over future states. Since it recovers normalized successor measure when $n \sim \mathrm{Geom}(1-\gamma)$, GHM can be seen as a special case of UHM. Furthermore, we can also represent single-step dynamics models with $n \sim \delta(1)$. Hence, UHM provides a unified framework for future prediction that subsumes both GHM and single-step dynamics models, as illustrated in Figure 1.

UHM can be learned from off-policy transitions via bootstrapping:

$$m^\pi(x \mid s, a, 1) = \mathcal{P}(x \mid s, a), \quad (9)$$

$$m^\pi(x \mid s, a, n+1) = \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a), a' \sim \pi(\cdot|s')} \left[m^\pi(x \mid s', a', n) \right], \quad (10)$$

analogous to the learning objective of GHM (4). Specifically, after sampling $n \sim p_H(\cdot)$ and $a' \sim \pi(\cdot|s')$, we train UHM so that its prediction for the $(n+1)$-step future state of (s, a) matches a bootstrapped sample $s_e \sim m^\pi(\cdot|s', a', n)$. Since the multi-step target is defined recursively through bootstrapping, learning the one-step case correctly provides the basis for learning the $n > 1$ future state distributions. Note that $m^\pi(x \mid s, a, n)$ is defined for each n, regardless of the horizon distribution p_H. Consequently, different p_H can be used throughout training or at inference time without altering the definition of m^π.

4.2. Critic Target Estimation

Based on the flexibility of UHM to represent and sample from arbitrary future distributions, we propose a generalized temporal difference learning framework.

Proposition 4.1. *For any sub-probability measure ν over $\mathbb{N}$, consider the ν-Bellman operator $\mathcal{T}^\nu$ defined as*

$$\mathcal{T}^\nu Q(s,a) \coloneqq \mathbb{E}\Big[R(s,a) + \gamma \sum_{k \geq 1} \big[\xi^\nu(k) R(s_k, a_k) + \nu(k) Q(s_k, a_k) \big] \;\Big|\; s_0 = s, a_0 = a, \pi \Big], \quad (11)$$

where

$$\xi^\nu(k) = \gamma^{k-1} - \sum_{\kappa=0}^{k-1} \left[\gamma^\kappa \nu(k-\kappa) \right]. \quad (12)$$

The iterative sequence $Q_{n+1} = \mathcal{T}^\nu Q_n$ from any bounded real function $Q_0 \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ converges to $Q^\pi(s,a)$.

A proof is provided in Appendix A.2. The TD learning with ν-Bellman operator recovers standard critic updates as special cases: choosing $\nu(k) = \gamma^{n-1}\mathbf{1}[k=n]$ yields n-step TD, while choosing $\nu(k) = (1-\lambda)(\lambda\gamma)^{k-1}$ represents TD(λ). To enable TD learning with the proposed framework (11) while avoiding repeated inference, we utilize UHM as it can sample states from the n-step future state distribution for any given n. Specifically, UHM allows one-sample estimation G^ν of the ν-Bellman backup target $\mathcal{T}^\nu Q(s,a)$ for any sub-probability measure ν as follows:

$$n \sim p_H, \quad (13)$$

$$s_e \sim m^\pi(\cdot|s,a,n), a_e \sim \pi(\cdot|s_e), \quad (14)$$

$$G^\nu = r + \gamma \left\{ w_\xi R(s_e, a_e) + w_\nu Q(s_e, a_e) \right\}, \quad (15)$$

where p_H is a horizon distribution over $\mathbb{N}$ satisfying $p_H(k) > 0$ whenever $\nu(k) \neq 0$ or $\xi^\nu(k) \neq 0$. The weights $w_\xi = \frac{\xi^\nu}{p_H}$ and $w_\nu = \frac{\nu}{p_H}$ are importance ratios that correct for the discrepancy between ν, ξ^ν and the sampling distribution p_H.

Among many possible choices of ν, we present a value learning method using the winsorized geometric measure, which is defined as follows:

$$\nu(k) = \begin{cases} (1-\lambda)(\lambda\gamma)^{k-1}, & \text{if } 1 \leq k < k_{\max}, \\ (\lambda\gamma)^{k_{\max}-1}, & \text{if } k = k_{\max}, \\ 0, & \text{if } k > k_{\max}, \end{cases} \quad (16)$$

which ensures the convergence guarantee in Proposition 4.1 since it is a sub-probability that satisfies $\sum_{k \geq 1} \nu(k) \leq 1$. The corresponding ξ^ν (12) yields

$$\xi^\nu(k) = \begin{cases} \lambda(\lambda\gamma)^{k-1}, & \text{if } 1 \leq k < k_{\max}, \\ 0, & \text{if } k \geq k_{\max}. \end{cases} \quad (17)$$

This choice retains the original form of TD(λ) while clipping excessively large and rare horizons, which in turn stabilizes the learning process. For the horizon distribution p_H,

we use the winsorized geometric distribution corresponding to the winsorized geometric measure (16):

$$n' \sim \text{Geom}(1-\lambda\gamma), n = \min(n', k_{\max}). \tag{18}$$

Then, we can obtain the value learning target G^ν (15) by substituting importance ratios:

$$w_\xi = \begin{cases} \frac{\lambda}{1-\lambda\gamma}, & \text{if } 1 \le n < k_{\max}, \\ 0, & \text{if } n = k_{\max}, \end{cases} \tag{19}$$

$$w_\nu = \begin{cases} \frac{1-\lambda}{1-\lambda\gamma}, & \text{if } 1 \le n < k_{\max}, \\ 1, & \text{if } n = k_{\max}. \end{cases} \tag{20}$$

In the following section, we provide a practical offline RL algorithm using this value learning target.

4.3. Practical Implementations

We now present a practical offline RL algorithm using UHM, whose pseudocode is provided in Algorithm 1. For notation, θ denotes the parameters of all neural networks and $\bar{\theta}$ represent their exponential moving average (EMA). At each update step, we perform the following update based on transition tuples (s, a, r, s', a') sampled from the dataset.

λ scheduling. Since the UHM is trained via bootstrapping, its output for large n is inaccurate in the early training stage. To resolve this issue, we perform scheduling as $\lambda = \frac{r\lambda_f}{1-(1-r)\lambda_f}$, where λ_f is the final trace value and $r \in [0, 1]$ is a training progress. It enforces the effective horizon of UHM to increase linearly from 1 to $1/(1-\lambda_f\gamma)$, making the bootstrapping of UHM more stable. Based on the scheduled λ, the maximum imaginary horizon $k_{\max}$ is decided as $\text{qgeom}(1-\lambda\gamma, q)$, which is a q-quantile of the geometric distribution $\text{Geom}(1-\lambda\gamma)$. As a result, the horizon n is sampled from the winsorized geometric distribution, which we denote as $\min(\text{Geom}(1-\lambda\gamma), k_{\max})$.

Model learning. We train a vector field v_θ to utilize flow-matching (Lipman et al., 2023) as a generative model for UHM, and follow coupled-CFM (Farebrother et al., 2025) to construct the learning objective (10). If $n > 1$, we first sample noise $s_e^0 \sim \mathcal{N}(0, I)$ and next action $\tilde{a}'$ to predict an $n-1$ step future state s_e^1, which is obtained by solving discretized ODE $s_e^{\tau+\Delta\tau} = x^\tau + v_{\bar{\theta}}(s_e^\tau | s', \tilde{a}', n-1, \tau)\Delta\tau$ from $\tau = 0$ to $\tau = 1$. Then s_e^0 is reused for constructing conditional optimal transport paths $s_e^\tau = (1-\tau)s_e^0 + \tau s_e^1$ for flow timestep $\tau \sim \text{Unif}[0, 1]$, resulting in the flow-matching loss $L^v = ||v_\theta(s_e^\tau | s, a, n, \tau) - (s_e^1 - s_e^0)||_2^2$. If $n = 1$, we skip solving the ODE and set $s_e^1 = s'$. We note that EMA weights $\bar{\theta}$ are used for generating bootstrapping targets to stabilize the training.

Behavior mixing. While UHM alleviates error accumulation through direct future prediction, it can still be inaccurate when queried with unseen state–action pairs. To address this issue, we introduce a simple behavior mixing strategy that uses a dataset action a' with probability β when generating the bootstrapping target s_e. Specifically, we deploy stochastically mixed policy $\pi^{\text{mix}} = (1-\beta)\pi_\theta + \beta\delta(a')$ to sample the next actions $\tilde{a}'$. It limits the total variation divergence from the behavior policy, analogous to classical RL papers (Kakade & Langford, 2002; Ross & Bagnell, 2010). We observe that the optimal choice of β varies across tasks; however, to avoid exhaustive hyperparameter tuning, we fix $\beta = 0.3$ in all main experiments. The detailed analysis on the effect of β is provided in Section 5.2.

Rewards and terminations. For reward modeling, we train a neural network r_θ by minimizing the mean-squared error $L^R = (R_\theta(s, a) - r)^2$. To properly handle terminal states, we employ an augmented state representation that concatenates a terminal indicator with the state, and train UHM to sample from this augmented space. We further treat terminal states as absorbing states that only transition to themselves and yield zero reward. It stabilizes value learning by prevent-

Algorithm 1 Offline RL with UHM

Input: UHM vector field v_θ, actor μ_θ, critic Q_θ, reward model R_θ, offline dataset $\mathcal{D}$, discount factor γ, actor noise scale σ, actor BC coefficient α, behavior mixing coefficient β, EMA decay η
$\bar{\theta} \leftarrow \theta$
for $i = 1$ **to** N_{update} **do**
  $(s, a, r, s', a') \sim \mathcal{D}$
  Schedule λ and $k_{\max}$
  // Sample future states for bootstrapping
  $n \sim \min(\text{Geom}(1-\lambda\gamma), k_{\max})$
  $\tilde{a}' \sim (1-\beta)\mathcal{N}(\mu_{\bar{\theta}}(s'), \sigma^2 I) + \beta\delta(a')$
  $s_e^0 \sim \mathcal{N}(0, I)$, $s_e^1 \leftarrow s_e^0$
  for $j = 1$ **to** N_{flow} **do**
    $s_e^1 \leftarrow s_e^1 + \frac{1}{N_{\text{flow}}} v_{\bar{\theta}}(s_e^1 | s', \tilde{a}', n-1, \frac{j-1}{N_{\text{flow}}})$
  end for
  if $n = 1$ **then** $s_e^1 \leftarrow s'$
  // UHM loss
  $\tau \sim \text{Unif}[0, 1]$, $s_e^\tau \leftarrow (1-\tau)s_e^0 + \tau s_e^1$
  $L^v \leftarrow ||v_\theta(s_e^\tau | s, a, n, \tau) - (s_e^1 - s_e^0)||^2$
  $a_e \sim \mathcal{N}(\mu_{\bar{\theta}}(s_e^1), \sigma^2 I)$
  // Actor-critic loss
  Compute w_ξ, w_ν according to (19)
  $G^\nu \leftarrow r + \gamma(w_\xi R_{\text{sg}(\theta)}(s_e^1, a_e) + w_\nu Q_{\bar{\theta}}(s_e^1, a_e))$
  $L^Q \leftarrow (Q_\theta(s, a) - G^\nu)^2$
  $L^\pi \leftarrow \alpha||\mu_\theta(s) - a||_2^2 - Q_{\text{sg}(\theta)}(s, \mu_\theta(s))$
  // Reward loss
  $L^R \leftarrow (R_\theta(s, a) - r)^2$
  // Update network parameters
  $L \leftarrow L^v + L^Q + L^R + L^\pi$
  $\theta \leftarrow \theta - \nabla_\theta L$, $\bar{\theta} \leftarrow (1-\eta)\bar{\theta} + \eta\theta$
end for

ing UHM from generating unseen combinations of states and terminal indicators. For notational simplicity, we do not introduce a separate symbol for the augmented state in Algorithm 1; however, explicitly modeling terminations and preventing value bootstrapping at terminal states are crucial for performance, as demonstrated in Section 5.2.

Actor-critic learning. We train a critic network Q_θ to minimize the TD learning objective $L^Q = (Q_\theta(s,a) - G^\nu)^2$, where G^ν is computed according to Equation (15) using the target network $Q_{\bar{\theta}}$. For actor learning, we train a deterministic actor network μ_θ to minimize the TD3+BC (Fujimoto & Gu, 2021) objective $L^\pi = \alpha\|\mu_\theta(s) - a\|_2^2 - Q_{\text{sg}(\theta)}(s, \mu_\theta(s))$, where $\text{sg}(\cdot)$ denotes the stop-gradient operator. During training, we apply target smoothing by adding Gaussian noise with standard deviation σ, resulting in a stochastic policy $\pi_\theta(\cdot \mid s) = \mathcal{N}(\mu_\theta(s), \sigma^2 I)$.

5. Experiments

We conduct extensive experiments to evaluate the proposed offline RL algorithm. In Section 5.1, we benchmark our method against competitive offline RL baselines across a diverse set of reward-based tasks in OGBench (Park et al., 2025a), including tasks that provide highly suboptimal datasets or require long-horizon reasoning. In Section 5.2, we conduct a series of analyses to examine how individual components of the proposed method contribute to overall performance.

5.1. Experiments on Offline RL Benchmarks

To evaluate the performance of the proposed method, we first conduct experiments on standard OGBench tasks that are commonly used in prior work (Park et al., 2025c; Li et al., 2025; Chen et al., 2025). We then study more challenging settings to examine the limits of our approach, including (1) noisy tasks with highly suboptimal datasets and (2) long-horizon reasoning tasks that require substantially more interaction steps to solve. For readability, we omit the "-singletask" suffix and shorten task names by using task indices (e.g., `cube-single-play-singletask-task1-v0` is denoted as `cube-single-play-1`).

5.1.1. Experimental Setup

Baselines. We compare the proposed method with several baselines. For model-free methods, we use *IQL* (Kostrikov et al., 2022) that trains a critic with in-sample maximization, *ReBRAC* (Tarasov et al., 2023) that performs behavior-regularization for actor and critic updates, and *FQL* (Park et al., 2025c) that trains a one-step flow policy regularized with behavior flow-matching policy. For model-based methods, we use *MOPO* (Yu et al., 2020) that penalizes rewards in predicted states with dynamics uncertainty, *MOBILE* (Sun et al., 2023) that replaces dynamics uncertainty in MOPO with a disagreement in critic targets, and *MAC* (Park et al., 2026a) that utilizes action chunking to prevent error accumulation and apply TD-n with synthetic rollouts.

Additional baselines. We also compare our method with additional baselines, which are designed for rigorous ablation studies. We first tune actor BC coefficient α for ReBRAC again and denote it *ReBRAC*†. Since it performs the same policy extraction method as our methods, we can examine whether our critic learning with UHM is effective by comparing with it. To assess the advantage of UHM over single-step dynamics models, we use *MBTD(λ)*, which generates synthetic rollouts using a single-step flow dynamics model to perform TD(λ) analogous to LEQ (Park & Lee, 2025). We also compare against *DTD(λ)*, which performs TD(λ) over trajectories sampled directly from the dataset, to determine whether on-policy value learning is necessary for performance. Finally, we compare our method with *GHM* that trains a critic with γ-MVE, to verify whether the increased flexibility of UHM provides tangible benefits in the offline RL setting. For fair comparison, all additional baselines use the same hyperparameters and design choices as the proposed method. Only actor BC coefficient α is tuned separately for each method.

Evaluation. For the standard OGBench tasks, we report the performance of model-free baselines from Park et al. (2025c) and model-based baselines from Park et al. (2026a). We additionally tune the model-based baselines for locomotion tasks following the same experimental protocol, as it is omitted in the paper. Each experiment is run with five random seeds, and we report the average success rate along with the standard deviation. All agents share the same network architectures and are trained for a total of 1M gradient steps. Performance is evaluated by averaging the results over the last three evaluation epochs. For both the proposed method and the additional baselines, the final value of the trace parameter λ_f is set to 0.8 and the discount factor is set to 0.999.

For noisy and long-horizon reasoning tasks, we report results only for the proposed method and the additional baselines, as these methods consistently outperform other approaches on the standard tasks. For noisy tasks, we follow the same evaluation protocol as in the standard setting. For long-horizon tasks, we use three random seeds and train agents for 2M gradient steps, while increasing λ_f to 0.9 to encourage horizon reduction. The dataset for the long-horizon reasoning tasks consists of 10M transitions, which is ten times larger than that used for the standard and noisy tasks. Please refer to Appendix B for more detailed experimental setup.

Table 1. Results on 50 standard tasks in OGBench. We omit the standard deviations of average success rates for methods whose results are taken from prior works (Park et al., 2025c; 2026a). Please refer to Table 7 for the detailed per-task results.

	Model-Free			Model-Based			Ablations				Ours
Environments (5 tasks each)	IQL	ReBRAC	FQL	MOPO	MOBILE	MAC	ReBRAC†	MBTD(λ)	DTD(λ)	GHM	UHM
antmaze-large-navigate	53 ± 3	81 ± 5	79 ± 3	0 ± 0	0 ± 0	18 ± 4	72 ± 14	71 ± 11	**93** ± 1	90 ± 2	89 ± 1
antmaze-giant-navigate	4 ± 1	26 ± 8	9 ± 5	0 ± 0	0 ± 0	0 ± 0	30 ± 13	27 ± 3	**52** ± 11	33 ± 7	36 ± 4
humanoidmaze-medium-navigate	33 ± 2	22 ± 8	58 ± 5	0 ± 0	0 ± 0	2 ± 0	22 ± 10	64 ± 11	81 ± 2	90 ± 1	**95** ± 1
humanoidmaze-large-navigate	2 ± 1	2 ± 1	4 ± 2	0 ± 0	0 ± 0	0 ± 0	1 ± 1	16 ± 3	27 ± 10	16 ± 1	**33** ± 9
antsoccer-arena-navigate	8 ± 2	0 ± 0	**60** ± 2	0 ± 0	0 ± 0	29 ± 4	1 ± 1	47 ± 3	0 ± 0	20 ± 4	26 ± 4
cube-single-play	83 ± 3	91 ± 2	96 ± 1	12 ± 4	81 ± 8	**99** ± 2	91 ± 2	92 ± 1	90 ± 2	91 ± 3	92 ± 3
cube-double-play	7 ± 1	12 ± 1	29 ± 2	1 ± 1	1 ± 2	**53** ± 4	4 ± 2	4 ± 1	4 ± 1	29 ± 1	30 ± 2
scene-play	28 ± 1	41 ± 3	56 ± 2	6 ± 8	8 ± 4	**97** ± 4	40 ± 3	31 ± 2	76 ± 3	44 ± 3	43 ± 4
puzzle-3x3-play	9 ± 1	21 ± 1	30 ± 1	20 ± 0	12 ± 9	20 ± 0	90 ± 4	93 ± 2	**99** ± 0	51 ± 2	**99** ± 1
puzzle-4x4-play	7 ± 1	14 ± 1	17 ± 2	0 ± 0	0 ± 0	**78** ± 13	1 ± 0	4 ± 0	1 ± 0	13 ± 1	11 ± 2
Average	23	31	44	4	10	40	35 ± 2	45 ± 2	52 ± 1	48 ± 1	**55** ± 1

Table 2. Results on 25 noisy tasks in OGBench. Please refer to Table 8 for the detailed per-task results.

Environments (5 tasks each)	ReBRAC†	MBTD(λ)	DTD(λ)	GHM	UHM
antmaze-medium-explore	74 ± 5	**96** ± 2	81 ± 6	91 ± 3	89 ± 4
antmaze-large-explore	13 ± 7	20 ± 2	18 ± 9	14 ± 2	**26** ± 5
cube-double-noisy	2 ± 1	2 ± 1	2 ± 2	**24** ± 3	18 ± 1
scene-noisy	27 ± 7	42 ± 2	6 ± 3	57 ± 3	**61** ± 3
puzzle-4x4-noisy	0 ± 0	0 ± 0	0 ± 0	**4** ± 1	1 ± 0
Average	23 ± 2	32 ± 1	23 ± 4	38 ± 1	**39** ± 1

Table 3. Results on 25 long-horizon reasoning tasks in OGBench. Please refer to Table 9 for the detailed per-task results.

Environments (5 tasks each)	ReBRAC†	MBTD(λ)	DTD(λ)	GHM	UHM
cube-triple-play	4 ± 1	12 ± 2	1 ± 1	44 ± 5	**56** ± 3
cube-quadruple-play	0 ± 0	0 ± 0	0 ± 1	12 ± 4	**16** ± 6
puzzle-4x5-play	7 ± 2	5 ± 3	14 ± 3	**17** ± 1	16 ± 0
puzzle-4x6-play	**11** ± 5	1 ± 1	5 ± 6	5 ± 3	**11** ± 1
humanoidmaze-giant-navigate	2 ± 1	5 ± 1	**46** ± 5	4 ± 0	10 ± 2
Average	5 ± 1	5 ± 1	13 ± 3	16 ± 2	**22** ± 1

5.1.2. Results

We now report experimental results across the three task categories. In all tables, the best-performing baseline is highlighted in bold. We additionally underline results that achieve at least 95% of the best success rate.

Results on standard tasks. Table 1 reports the mean success rates and standard deviations across the standard OGBench tasks. Overall, model-based baselines exhibit weaker performance than model-free baselines across most tasks. In particular, MOPO and MOBILE achieve success rates below 10% on nearly all tasks, suggesting that uncertainty-based reward penalties may significantly hinder value learning in sparse-reward settings. MAC significantly outperforms other baselines on several manipulation tasks, but performs poorly on locomotion tasks. We hypothesize that this limitation arises from rejection sampling over the behavior policy, which may struggle to handle high-dimensional actions.

Among the additional baselines, methods that leverage n-step returns for TD learning outperform one-step TD methods, such as ReBRAC† and FQL. This highlights the importance of horizon reduction in value learning. Notably, although MBTD(λ) and GHM perform theoretically sound value learning via model-based value expansion, they achieve lower performance than DTD(λ), which uses dataset trajectories. In contrast to these approaches, UHM is the only method that outperforms DTD(λ). We attribute this advantage to directly predicting n-step future states while capping the maximum future horizon, which makes value learning more scalable and robust.

Results on noisy tasks. In Table 2, we report the mean and standard deviations of success rates across 25 noisy tasks in OGBench. Compared to standard tasks, DTD(λ) struggles to learn effective policies and even exhibits comparable average success rates with ReBRAC†. It achieves success rates below 10% on all manipulation environments, which is substantially lower than its performance on standard tasks. This observation suggests that relying on trajectories from highly suboptimal datasets can severely hinder effective value learning. On the other hand, model-based approaches consistently outperform model-free methods on noisy tasks. We attribute this improvement to the use of imagined on-policy trajectories rather than directly relying on suboptimal transitions from the dataset. Among model-based approaches, GHM and UHM outperform MBTD(λ), with UHM achieving the best overall performance. These results demonstrate the scalability of the proposed method to offline RL tasks with highly suboptimal data.

Results on long-horizon reasoning tasks. In Table 3, we report performance on long-horizon reasoning tasks in OGBench. Both GHM and UHM achieve stronger performance than DTD(λ), suggesting that relying on dataset trajectories for long-horizon returns can lead to increased distributional mismatch. MBTD(λ) performs comparably to ReBRAC†, failing to yield meaningful gains from long-horizon imagination when using single-step dynamics models. Among the compared methods, UHM achieves the best overall performance, improving the average success rate by approximately 69% over DTD(λ) and 38% over GHM, demonstrating its scalability to long-horizon reasoning tasks. How-

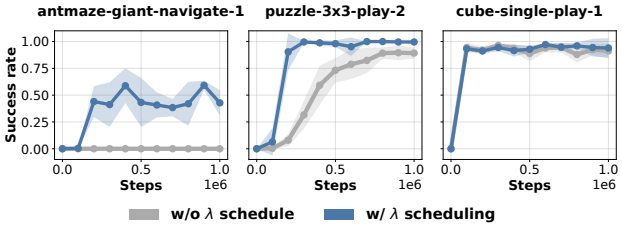

Figure 2. Learning curves with and without λ scheduling.

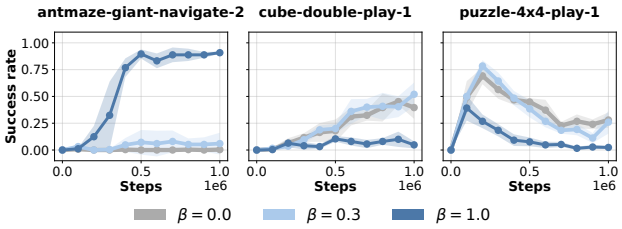

Figure 3. Learning curves for different behavior mixing coefficients β.

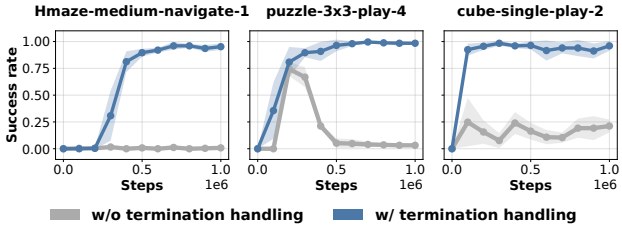

Figure 4. Learning curves with and without terminal state handling.

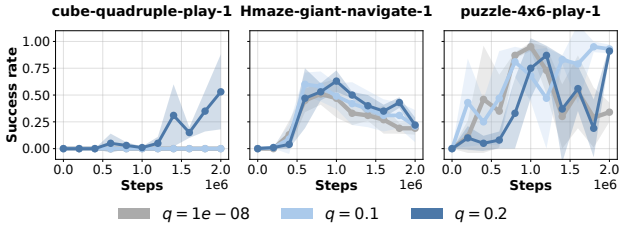

Figure 5. Learning curves for different horizon winsorization quantiles q.

ever, in `humanoidmaze-giant`, DTD(λ) significantly outperforms model-based methods, suggesting that accurate model learning remains challenging in large-scale, high-dimensional domains. This result highlights the need for future research that combines our approach with techniques for handling high-dimensional observations and for reducing the effective decision horizon of the policy.

To summarize, our experimental results reveal three key findings. First, horizon reduction plays a crucial role in effective value learning. Second, model-based value expansion without repeated inference scales well to noisy tasks and long-horizon reasoning tasks. Finally, the proposed method shows competitive performance across the task categories we consider, with a 14% higher average success rate than the second-best method.

5.2. Ablation Studies

In this section, we provide ablation studies to analyze the contribution of each component of the proposed method. In all figures, shaded areas denote a standard deviation over five independent runs.

Ablations on λ scheduling. Figure 2 compares the learning curves of the proposed method with and without λ scheduling. Empirically, λ scheduling consistently improves performance across nearly all tasks. In `antmaze-giant-navigate-1`, the method fails to learn meaningful policies without scheduling, whereas introducing λ scheduling enables better performance. Even in relatively easier tasks, λ scheduling yields consistent performance gains and faster convergence. These results suggest that λ scheduling enables efficient value learning.

Ablations on the behavior mixing coefficient β. Figure 3 illustrates the effect of the behavior mixing coefficient β. We observe that the optimal value of β varies across tasks. In `antmaze-giant-navigate-2`, setting $\beta = 1.0$ is crucial for stable learning, whereas smaller values of β lead to poor performance. In contrast, tasks such as `cube-double-play-1` and `puzzle-4x4-play-1` benefit from smaller values of β. Meanwhile, performance averaged over all representative tasks from the standard environments remains relatively robust to the choice of β, with success rates of 0.63, 0.66, and 0.59 for $\beta = 0.0, 0.3$, and 1.0, respectively. Based on these observations, we recommend tuning β to achieve optimal task-specific performance, while we fix $\beta = 0.3$ in our main experiments for simplicity.

Ablations on terminal state handling. Figure 4 illustrates the effect of explicitly handling terminal states. Ignoring terminal states leads to a substantial degradation in performance across all evaluated tasks. We can infer that bootstrapping value estimates from terminal states causes unstable learning. These results highlight that handling terminal states is crucial for the performance.

Ablations on the horizon winsorization quantile q. Figure 5 illustrates the effect of the winsorization quantile q on a subset of long-horizon reasoning tasks. Overall, $q = 0.1$ and $q = 0.2$ outperform $q = 10^{-8}$, indicating that winsorization is important for performance. However, the best choice of q can be task-dependent: `cube-quadruple-play-1` performs well only with $q = 0.2$, while `puzzle-4x6-play-1` is much more stable with $q = 0.1$. We therefore fix $q = 0.2$ in the main experiments for simplicity, although tuning q may further improve performance.

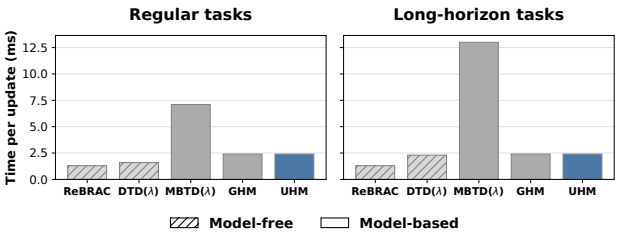

Figure 6. Wall-clock time per gradient update across baselines.

Update time comparison. Figure 6 compares the wall-clock time required for a single gradient update with an RTX 4090 GPU. While model-free approaches demonstrate the lowest update time, MBTD(λ) exhibits substantially higher update time, as it relies on repeated inference of a dynamics model. Notably, GHM and UHM significantly reduce this overhead by directly predicting future states. This advantage is more pronounced in long-horizon tasks, where UHM achieves update times within 10% of DTD(λ) while retaining the benefits of model-based value expansion.

6. Conclusion

In this work, we introduce universal horizon models (UHM), which generalize geometric horizon models by allowing the future horizon to be sampled from arbitrary distributions. We further propose a scalable algorithm for offline model-based value expansion and demonstrate that our method outperforms baselines across diverse tasks in OGBench. Despite these results, our approach has several limitations. First, the accuracy of UHM is constrained by data scarcity, which can lead the model to extrapolate beyond the data support. It suggests the need for additional mechanisms to guide predictions toward in-distribution states. Second, the performance of UHM is limited by model capacity, as accurately modeling longer horizons may require more expressive network architectures. Addressing these limitations and extending UHM to handle visual observations with action chunks remain interesting directions for future work.

Acknowledgments

This work was partly supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-01190, [SW Star Lab] Robot Learning: Efficient, Safe, and Socially-Acceptable Machine Learning, 50%), and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (NO.RS-2021-II211343, Artificial Intelligence Graduate School Program [Seoul National University], 50%).

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none of which we feel must be specifically highlighted here.

References

Ajay, A., Du, Y., Gupta, A., Tenenbaum, J., Jaakkola, T., and Agrawal, P. Is conditional generative modeling all you need for decision-making? In *Proceedings of the International Conference on Learning Representations*, 2023.

An, G., Moon, S., Kim, J.-H., and Song, H. O. Uncertainty-based offline reinforcement learning with diversified q-ensemble. In *Advances in Neural Information Processing Systems*, 2021.

Blier, L., Tallec, C., and Ollivier, Y. Learning successor states and goal-dependent values: A mathematical viewpoint. *arXiv preprint arXiv:2101.07123*, 2021.

Chen, D., Liu, Y., Zhou, Z., Qu, C., and Qi, Y. Unleashing flow policies with distributional critics. *arXiv preprint arXiv:2509.23087*, 2025.

Chen, H., Lu, C., Ying, C., Su, H., and Zhu, J. Offline reinforcement learning via high-fidelity generative behavior modeling. In *Proceedings of the International Conference on Learning Representations*, 2023.

Cheng, J., Qiao, R., Ma, Y., Li, B., Xiong, G., Miao, Q., Li, Y., and Lv, Y. Scaling offline model-based RL via jointly-optimized world-action model pretraining. In *Proceedings of the International Conference on Learning Representations*, 2025.

Dayan, P. Improving generalization for temporal difference learning: The successor representation. *Neural computation*, 5(4):613–624, 1993.

De Asis, K., Hernandez-Garcia, J., Holland, G., and Sutton, R. Multi-step reinforcement learning: A unifying algorithm. In *Proceedings of the AAAI conference on artificial intelligence*, 2018.

Farebrother, J., Pirotta, M., Tirinzoni, A., Munos, R., Lazaric, A., and Touati, A. Temporal difference flows. In *Proceedings of the International Conference on Machine Learning*, 2025.

Feinberg, V., Wan, A., Stoica, I., Jordan, M. I., Gonzalez, J. E., and Levine, S. Model-based value expansion for efficient model-free reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2018.

Fujimoto, S. and Gu, S. S. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2021.

Garg, D., Hejna, J., Geist, M., and Ermon, S. Extreme Q-learning: Maxent RL without entropy. In *Proceedings of the International Conference on Learning Representations*, 2023.

Ghasemipour, K., Gu, S. S., and Nachum, O. Why so pessimistic? Estimating uncertainties for offline RL through ensembles, and why their independence matters. In *Advances in Neural Information Processing Systems*, 2022.

Hafner, D., Pasukonis, J., Ba, J., and Lillicrap, T. Mastering diverse control tasks through world models. *Nature*, pp. 1–7, 2025.

Hansen, N., Su, H., and Wang, X. TD-MPC2: Scalable, robust world models for continuous control. In *Proceedings of the International Conference on Learning Representations*, 2024.

Hansen-Estruch, P., Kostrikov, I., Janner, M., Kuba, J. G., and Levine, S. IDQL: Implicit Q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*, 2023.

Hendrycks, D. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

Hernandez-Garcia, J. F. and Sutton, R. S. Understanding multi-step deep reinforcement learning: A systematic study of the DQN target. *arXiv preprint arXiv:1901.07510*, 2019.

Hessel, M., Modayil, J., Van Hasselt, H., Schaul, T., Ostrovski, G., Dabney, W., Horgan, D., Piot, B., Azar, M., and Silver, D. Rainbow: Combining improvements in deep reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, 2018.

Hong, M., Kang, M., and Oh, S. Diffused task-agnostic milestone planner. In *Advances in Neural Information Processing Systems*, 2023.

Jackson, M. T., Matthews, M. T., Lu, C., Ellis, B., Whiteson, S., and Foerster, J. Policy-guided diffusion. In *Proceedings of the Reinforcement Learning Conference*, 2024.

Janner, M., Fu, J., Zhang, M., and Levine, S. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, 2019.

Janner, M., Mordatch, I., and Levine, S. γ-models: Generative temporal difference learning for infinite-horizon prediction. In *Advances in Neural Information Processing Systems*, 2020.

Janner, M., Li, Q., and Levine, S. Offline reinforcement learning as one big sequence modeling problem. In *Advances in Neural Information Processing Systems*, 2021.

Janner, M., Du, Y., Tenenbaum, J., and Levine, S. Planning with diffusion for flexible behavior synthesis. In *Proceedings of the International Conference on Machine Learning*, 2022.

Kakade, S. and Langford, J. Approximately optimal approximate reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2002.

Kidambi, R., Rajeswaran, A., Netrapalli, P., and Joachims, T. MOReL: Model-based offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020.

Kingma, D. P. Adam: A method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations*, 2015.

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit Q-learning. In *Proceedings of the International Conference on Learning Representations*, 2022.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative Q-learning for offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2020.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Li, Q., Zhou, Z., and Levine, S. Reinforcement learning with action chunking. In *Advances in Neural Information Processing Systems*, 2025.

Li, Q., Park, S., and Levine, S. Decoupled Q-chunking. In *Proceedings of the International Conference on Learning Representations*, 2026.

Lin, H., Xu, Y.-Y., Sun, Y., Zhang, Z., Li, Y.-C., Jia, C., Ye, J., Zhang, J., and Yu, Y. Any-step dynamics model improves future predictions for online and offline reinforcement learning. In *Proceedings of the International Conference on Learning Representations*, 2025.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *Proceedings of the International Conference on Learning Representations*, 2023.

Park, K. and Lee, Y. Model-based offline reinforcement learning with lower expectile Q-learning. In *Proceedings of the International Conference on Learning Representations*, 2025.

Park, K., Park, S., Lee, Y., and Levine, S. Scalable offline model-based RL with action chunks. In *Proceedings of the International Conference on Learning Representations*, 2026a.

Park, S., Frans, K., Eysenbach, B., and Levine, S. OGBench: Benchmarking offline goal-conditioned RL. In *Proceedings of the International Conference on Learning Representations*, 2025a.

Park, S., Frans, K., Mann, D., Eysenbach, B., Kumar, A., and Levine, S. Horizon reduction makes RL scalable. In *Advances in Neural Information Processing Systems*, 2025b.

Park, S., Li, Q., and Levine, S. Flow Q-learning. In *Proceedings of the International Conference on Machine Learning*, 2025c.

Park, S., Oberai, A., Atreya, P., and Levine, S. Transitive RL: Value learning via divide and conquer. In *Proceedings of the International Conference on Learning Representations*, 2026b.

Peng, X. B., Kumar, A., Zhang, G., and Levine, S. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Peters, J. and Schaal, S. Reinforcement learning by reward-weighted regression for operational space control. In *Proceedings of the International Conference on Machine Learning*, 2007.

Prudencio, R. F., Maximo, M. R., and Colombini, E. L. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE Transactions on Neural Networks and Learning Systems*, 35(8):10237–10257, 2023.

Rosete-Beas, E., Mees, O., Kalweit, G., Boedecker, J., and Burgard, W. Latent plans for task-agnostic offline reinforcement learning. In *Proceedings of the Conference on Robot Learning*, 2023.

Ross, S. and Bagnell, D. Efficient reductions for imitation learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2010.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. In *Proceedings of the International Conference on Learning Representations*, 2015.

Seo, Y. and Abbeel, P. Coarse-to-fine Q-network with action sequence for data-efficient reinforcement learning. In *Advances in Neural Information Processing Systems*, 2025.

Sikchi, H., Zheng, Q., Zhang, A., and Niekum, S. Dual RL: Unification and new methods for reinforcement and imitation learning. In *Proceedings of the International Conference on Learning Representations*, 2024.

Singh, A., Liu, H., Zhou, G., Yu, A., Rhinehart, N., and Levine, S. Parrot: Data-driven behavioral priors for reinforcement learning. In *Proceedings of the International Conference on Learning Representations*, 2021.

Sun, Y., Zhang, J., Jia, C., Lin, H., Ye, J., and Yu, Y. Model-bellman inconsistency for model-based offline reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2023.

Sutton, R. S., Barto, A. G., et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Talvitie, E. Model regularization for stable sample rollouts. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2014.

Tarasov, D., Kurenkov, V., Nikulin, A., and Kolesnikov, S. Revisiting the minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Thakoor, S., Rowland, M., Borsa, D., Dabney, W., Munos, R., and Barreto, A. Generalised policy improvement with geometric policy composition. In *Proceedings of the International Conference on Machine Learning*, 2022.

Wu, Y., Zhai, S., Srivastava, N., Susskind, J., Zhang, J., Salakhutdinov, R., and Goh, H. Uncertainty weighted actor-critic for offline reinforcement learning. In *Proceedings of the International Conference on Machine Learning*, 2021.

Xu, H., Jiang, L., Li, J., Yang, Z., Wang, Z., Chan, V. W. K., and Zhan, X. Offline RL with no OOD actions: In-sample learning via implicit value regularization. In *Proceedings of the International Conference on Learning Representations*, 2023.

Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J. Y., Levine, S., Finn, C., and Ma, T. MOPO: Model-based offline policy optimization. In *Advances in Neural Information Processing Systems*, 2020.

Zhang, J., Springenberg, J. T., Byravan, A., Hasenclever, L., Abdolmaleki, A., Rao, D., Heess, N., and Riedmiller, M. Leveraging jumpy models for planning and fast learning in robotic domains. *arXiv preprint arXiv:2302.12617*, 2023.

Zheng, C., Park, S., Levine, S., and Eysenbach, B. Intention-conditioned flow occupancy models. In *Proceedings of the International Conference on Learning Representations*, 2026.

A. Theoretical Analysis

A.1. Derivation of Equation 5

γ-MVE (Janner et al., 2020) is originally proposed to conduct value learning after a H-step GHM rollout:

$$Q_{\gamma\text{-MVE}} = R(s,a) + \frac{\gamma}{1-\gamma}\sum_{n=1}^{H}\frac{(1-\gamma)(\gamma-\tilde{\gamma})^{n-1}}{(1-\tilde{\gamma})^{n}}\mathbb{E}_{\substack{s_e\sim m_n^\pi(\cdot|s,a)\\ a_e\sim\pi(\cdot|s_e)}}\left[R(s_e,a_e)\right] + \gamma\left(\frac{\gamma-\tilde{\gamma}}{1-\tilde{\gamma}}\right)^{H}\mathbb{E}_{\substack{s_e\sim m_H^\pi(\cdot|s,a)\\ a_e\sim\pi(\cdot|s_e)}}\left[Q(s_e,a_e)\right], \tag{21}$$

where m_n^π denotes a distribution over states at the nth sequential step of a GHM rollout, and $\tilde{\gamma}\in(0,\gamma]$ denotes a discount factor to train the GHM. For the single-step rollout case, the equation (21) is simplified as follows:

$$Q_{\gamma\text{-MVE}} = R(s,a) + \gamma\mathbb{E}_{\substack{s_e\sim m^\pi(\cdot|s,a)\\ a_e\sim\pi(\cdot|s_e)}}\left[\frac{1}{1-\tilde{\gamma}}R(s_e,a_e) + \frac{\gamma-\tilde{\gamma}}{1-\tilde{\gamma}}Q(s_e,a_e)\right]. \tag{22}$$

By defining $\lambda := \tilde{\gamma}/\gamma$, the equation (22) can be rewritten as follows:

$$Q_{\gamma\text{-MVE}} = R(s,a) + \gamma\mathbb{E}_{\substack{s_e\sim m^\pi(\cdot|s,a)\\ a_e\sim\pi(\cdot|s_e)}}\left[\frac{1}{1-\lambda\gamma}R(s_e,a_e) + \frac{(1-\lambda)\gamma}{1-\lambda\gamma}Q(s_e,a_e)\right]. \tag{23}$$

Using the definition of GHM $m^\pi(s_e|s,a) = (1-\lambda\gamma)\sum_{k=0}^{\infty}(\lambda\gamma)^k\text{Pr}(s_{k+1}=s_e|s_0=s,a_0=a,\pi)$, we can derive that expectations of γ-MVE target and TD(λ) target are equal.

$$\begin{aligned}
Q_{\gamma\text{-MVE}} &= \sum_{k=0}^{\infty}\{R(s_e,a_e)+(1-\lambda)\gamma Q(s_e,a_e)\}(\lambda\gamma)^k\text{Pr}(s_{k+1}=s_e,a_k=a_e|s_0=s,a_0=a,\pi) && (24)\\
&= \mathbb{E}\left[\sum_{k=0}^{\infty}(\lambda\gamma)^k R_{k+1} + (1-\lambda)\gamma\sum_{k=0}^{\infty}(\lambda\gamma)^k Q(s_{k+1},a_{k+1}) \;\middle|\; s_0=s,a_0=a,\pi\right] && (25)\\
&= \mathbb{E}\left[(1-\lambda)\sum_{k=0}^{\infty}\lambda^k(\sum_{l=0}^{k}\gamma^l R_{l+1} + \gamma^{k+1}Q(s_{k+1},a_{k+1})) \;\middle|\; s_0=s,a_0=a,\pi\right] && (26)\\
&= \mathbb{E}\left[(1-\lambda)\sum_{k=0}^{\infty}\lambda^k G_t^{(k+1)} \;\middle|\; s_0=s,a_0=a,\pi\right] && (27)\\
&= \mathbb{E}\left[G_t^\lambda \;\middle|\; s_0=s,a_0=a,\pi\right]. && (28)
\end{aligned}$$

This result indicates that GHM realizes TD(λ) with single-step rollouts, avoiding the need for recursive inference inherent to methods based on single-step dynamics models.

A.2. Proof of Proposition 4.1

Define the Banach space $\mathcal{B}(\mathcal{S}\times\mathcal{A})$ as the set of all bounded functions $Q:\mathcal{S}\times\mathcal{A}\to\mathbb{R}$ equipped with the supremum norm $\|Q\|_\infty := \sup_{(s,a)\in\mathcal{S}\times\mathcal{A}}|Q(s,a)|$. Let ν be a sub-probability measure on $\mathbb{N}$, i.e., $\nu(k)\geq 0$ and $\sum_{k\geq 1}\nu(k)\leq 1$. Recall the ν-Bellman operator $\mathcal{T}^\nu$ defined by

$$(\mathcal{T}^\nu Q)(s,a) := \mathbb{E}\Big[R(s,a) + \gamma\sum_{k\geq 1}\left(\xi^\nu(k)R(s_k,a_k) + \nu(k)Q(s_k,a_k)\right) \;\Big|\; s_0=s,a_0=a,\pi\Big]. \tag{29}$$

For any $Q, Q' \in \mathcal{B}(\mathcal{S} \times \mathcal{A})$ and any $(s, a) \in \mathcal{S} \times \mathcal{A}$,

$$\big|(\mathcal{T}^\nu Q)(s,a) - (\mathcal{T}^\nu Q')(s,a)\big| = \gamma \Big|\mathbb{E}\Big[\sum_{k\geq 1} \nu(k)\big(Q(s_k,a_k) - Q'(s_k,a_k)\big) \,\Big|\, s_0 = s, a_0 = a, \pi\Big]\Big| \tag{30}$$

$$\leq \gamma\, \mathbb{E}\Big[\sum_{k\geq 1} \nu(k)\,\big|Q(s_k,a_k) - Q'(s_k,a_k)\big| \,\Big|\, s_0 = s, a_0 = a, \pi\Big] \tag{31}$$

$$\leq \gamma\Big(\sum_{k\geq 1} \nu(k)\Big)\|Q - Q'\|_\infty \tag{32}$$

$$\leq \gamma \|Q - Q'\|_\infty. \tag{33}$$

Taking the supremum over (s, a) yields

$$\|\mathcal{T}^\nu Q - \mathcal{T}^\nu Q'\|_\infty \leq \gamma \|Q - Q'\|_\infty, \tag{34}$$

so $\mathcal{T}^\nu$ is a γ-contraction. The Banach fixed point theorem implies that $\mathcal{T}^\nu$ admits a unique fixed point $Q^\star \in \mathcal{B}(\mathcal{S} \times \mathcal{A})$, and the iteration $Q_{n+1} = \mathcal{T}^\nu Q_n$ converges in $\|\cdot\|_\infty$ to $Q^\star$ for any initial $Q_0 \in \mathcal{B}(\mathcal{S} \times \mathcal{A})$.

Recall

$$Q^\pi(s,a) = \mathbb{E}\Big[\sum_{j\geq 0} \gamma^j R(s_j, a_j) \,\Big|\, s_0 = s, a_0 = a, \pi\Big]. \tag{35}$$

Substituting Q^π into $\mathcal{T}^\nu$ gives

$$(\mathcal{T}^\nu Q^\pi)(s,a) = \mathbb{E}\Big[R(s,a) + \gamma \sum_{k\geq 1} \xi^\nu(k) R(s_k,a_k) + \gamma \sum_{i\geq 1} \nu(i) Q^\pi(s_i,a_i) \,\Big|\, s_0 = s, a_0 = a, \pi\Big] \tag{36}$$

$$= \mathbb{E}\Big[R(s,a) + \gamma \sum_{k\geq 1} \xi^\nu(k) R(s_k,a_k) + \gamma \sum_{i\geq 1} \nu(i) \sum_{j\geq 0} \gamma^j R(s_{i+j}, a_{i+j}) \,\Big|\, s_0 = s, a_0 = a, \pi\Big]. \tag{37}$$

Re-index the double sum by $k = i + j$ (≥ 1):

$$\gamma \sum_{i\geq 1} \nu(i) \sum_{j\geq 0} \gamma^j R(s_{i+j}, a_{i+j}) = \gamma \sum_{k\geq 1} \left(\sum_{i=1}^{k} \gamma^{k-i} \nu(i)\right) R(s_k, a_k). \tag{38}$$

Therefore, for each $k \geq 1$, the coefficient of $R(s_k, a_k)$ equals

$$\gamma \xi^\nu(k) + \gamma \sum_{i=1}^{k} \gamma^{k-i} \nu(i) = \gamma \gamma^{k-1} = \gamma^k. \tag{39}$$

Hence,

$$(\mathcal{T}^\nu Q^\pi)(s,a) = \mathbb{E}\Big[R(s,a) + \sum_{k\geq 1} \gamma^k R(s_k,a_k) \,\Big|\, s_0 = s, a_0 = a, \pi\Big] \tag{40}$$

$$= \mathbb{E}\Big[\sum_{k\geq 0} \gamma^k R(s_k,a_k) \,\Big|\, s_0 = s, a_0 = a, \pi\Big] \tag{41}$$

$$= Q^\pi(s,a). \tag{42}$$

Thus Q^π is a fixed point of $\mathcal{T}^\nu$. By uniqueness of the fixed point, $Q^\star = Q^\pi$. Therefore, $\mathcal{T}^\nu$ converges to Q^π under repeated application, i.e., $Q_{n+1} = \mathcal{T}^\nu Q_n$ implies $Q_n \to Q^\pi$ in $\|\cdot\|_\infty$.

B. Experimental Details

B.1. Tasks

OGBench. All experiments are conducted on OGBench, a large-scale benchmark originally designed for offline goal-conditioned reinforcement learning. Following prior work, we use the single-task variants ("-singletask") of OGBench to benchmark standard reward-maximizing offline RL methods. Each OGBench environment provides five evaluation goals, corresponding to different tasks, where transitions in the dataset are relabeled with a semi-sparse reward function for a fixed goal. The episode terminates when the agent reaches the goal configuration. We refer readers to prior work for a detailed description of the benchmark and reward construction (Park et al., 2025a;c).

Our main experiments are conducted on three task sets, each designed to evaluate different challenges in offline RL.

Standard tasks. The first task set consists of 10 standard environments (50 tasks in total) commonly used in prior offline RL studies. This set includes five locomotion environments and five manipulation environments:

- **Locomotion**
  - `antmaze-large-navigate`
  - `antmaze-giant-navigate`
  - `humanoidmaze-medium-navigate`
  - `humanoidmaze-large-navigate`
  - `antsoccer-arena-navigate`
- **Manipulation**
  - `cube-single-play`
  - `cube-double-play`
  - `scene-play`
  - `puzzle-3x3-play`
  - `puzzle-4x4-play`

Noisy tasks. The second task set focuses on learning from highly suboptimal datasets, where effective policies must be recovered despite limited or noisy coverage. This set includes five environments (25 tasks in total) with exploratory or noisy data distributions:

- `antmaze-large-explore`
- `antmaze-giant-explore`
- `cube-double-play-noisy`
- `puzzle-4x4-play-noisy`
- `scene-play-noisy`

Long-horizon reasoning tasks. The third task set consists of long-horizon reasoning tasks, where reaching the target configuration requires significantly more environment steps compared to the previous task sets. This set includes five environments (25 tasks in total) as follows:

- `cube-triple-play`
- `cube-quadruple-play`
- `puzzle-4x5-play`
- `puzzle-4x6-play`
- `humanoidmaze-giant-navigate`

We note that `cube-octuple-play` is excluded since none of the baselines show meaningful results.

B.2. Baselines and Hyperparameters

In this section, we describe the baselines used in our main experiments along with their hyperparameter settings. Hyperparameters that are shared across methods and environments are reported in Table 4. For method-specific hyperparameters that require tuning, we follow the protocol of Park et al. (2025c) and tune each method on the default task of each environment. Below, we provide a brief description of each baseline and the method-specific hyperparameters that we tuned. For model-based baselines, we use the codebase of Park et al. (2026a).

- **IQL (Kostrikov et al., 2022)** trains a critic using expectile regression, which prevents querying out-of-distribution actions when computing TD targets. The policy is trained via advantage-weighted regression. We use the results reported in Park et al. (2025c).
- **ReBRAC (Tarasov et al., 2023)** performs SARSA-style updates while regularizing the training objective with the mean-squared error between behavior actions and actions sampled from the current policy. We use the results reported in Park et al. (2025c).
- **FQL (Park et al., 2025c)** trains a one-step push-forward policy regularized by a behavior flow-matching objective, and updates the critic using SARSA. We use the results reported in Park et al. (2025c).
- **MOPO (Yu et al., 2020)** generates synthetic transitions using learned dynamics models and penalizes the rewards of synthetic states based on model uncertainty. For standard manipulation tasks, we use the results reported in Park et al. (2026a). For locomotion tasks, we sweep over penalty coefficients $\omega \in \{0.1, 0.5, 1.0, 2.0, 3.0, 5.0\}$, and fix the rollout length $H = 10$ and the model batch ratio $f = 0.25$. Since MOPO consistently yields near-zero success rates on locomotion tasks, we do not report individual values.
- **MOBILE (Sun et al., 2023)** replaces the dynamics uncertainty penalty in MOPO with uncertainty estimates in the Bellman backup targets. Hyperparameter tuning follows the same protocol as MOPO. However, similar to MOPO, it also achieves zero reward on every locomotion environment, so we do not report the environment-specific values.
- **MAC (Park et al., 2026a)** employs a flow-based action-chunking policy and generates long-horizon rollouts by deploying it over learned dynamics models. For standard manipulation tasks, we use the results reported in Park et al. (2026a). For locomotion tasks, we follow the default hyperparameter settings provided in the paper.

We now present an explanation of additional baselines and the method-specific hyperparameters. Our implementation of additional baselines and the proposed method is based on the codebase of Park et al. (2025c).

- **ReBRAC†** is tuned under our experimental setup, using the common hyperparameters reported in Table 4. Compared to Park et al. (2025c), we use a larger discount factor $\gamma = 0.999$ for all environments. We only tune the actor BC coefficient α separately for each environment, and fix the critic BC coefficient to zero as it has a relatively marginal effect.
- **MBTD(λ)** is a model-based approach that generates n-step rollouts using a single-step dynamics model and performs TD(λ) on the generated trajectories. For controlled ablation studies, we model the dynamics using flow-matching with the same number of flow steps as our method. We also apply the same techniques used in our approach, including λ scheduling, terminal state handling, and horizon winsorization. We only select the behavior-cloning coefficient α separately for each environment.
- **DTD(λ)** is a model-free method that performs TD(λ) on trajectories sampled directly from the dataset. As with MBTD(λ), we apply the same techniques as our method, including λ scheduling, terminal state handling, and horizon winsorization. We only select the behavior-cloning coefficient α separately for each environment.
- **GHM** is a model-based method that uses geometric horizon models to predict discounted future states and performs γ-MVE. It is identical to our method except that the horizon n is not provided as an input to the model and horizon winsorization is not applicable. We only select the behavior-cloning coefficient α separately for each environment.

To ensure a fair comparison, the proposed method and additional baselines share the same design choices, such as λ scheduling, and hyperparameters reported in Table 5. The only hyperparameter we tuned is the actor BC coefficient α. It is selected based on the performance of DTD(λ) for each environment from the candidate set $\{0.003, 0.01, 0.03, 0.1, 0.3, 1.0\}$. The selected BC coefficients for each environment are reported in Table 6.

Table 4. Common hyperparameters for offline RL experiment.

Hyperparameter	**Value**
Dataset size	1M (default), 10M (long-horizon reasoning tasks)
Learning rate	0.0003
Optimizer	Adam (Kingma, 2015)
Gradient steps	1M (default), 2M (long-horizon reasoning tasks)
Minibatch size	256
MLP dimensions	[512, 512, 512, 512]
Nonlinearity	GELU (Hendrycks, 2016)
EMA decay η	0.005

Table 5. Common hyperparameters of the proposed method and additional baselines.

Hyperparameter	**Value**
Discount factor γ	0.999
Clipped double Q-learning	True (locomotion tasks), False (manipulation tasks)
Flow steps	5
ODE solver	Midpoint
Final td-lambda λ_f	0.8 (default), 0.9 (long-horizon reasoning tasks)
Behavior mixing coefficient β	0.3
Winsorization quantile q	0.2

Table 6. BC coefficient α used in main experiments.

	ReBRAC†	MBTD(λ)	DTD(λ)	GHM	UHM
antmaze-large-navigate	0.01	0.01	0.01	0.01	0.01
antmaze-giant-navigate	0.01	0.01	0.01	0.01	0.01
humanoidmaze-medium-navigate	0.01	0.01	0.01	0.01	0.01
humanoidmaze-large-navigate	0.01	0.01	0.01	0.01	0.01
antsoccer-arena-navigate	0.01	0.01	0.01	0.01	0.01
cube-single-play	1.0	1.0	1.0	1.0	1.0
cube-double-play	0.1	0.1	0.1	0.1	0.1
scene-play	0.1	0.1	0.1	0.1	0.1
puzzle-3x3-play	0.3	0.3	0.3	0.3	0.3
puzzle-4x4-play	0.1	0.1	0.1	0.1	0.1
antmaze-medium-explore	0.003	0.01	0.003	0.01	0.01
antmaze-large-explore	0.003	0.01	0.003	0.01	0.01
cube-double-noisy	0.01	0.01	0.01	0.01	0.01
scene-noisy	0.03	0.03	0.03	0.03	0.03
puzzle-4x4-noisy	0.01	0.01	0.01	0.01	0.01
cube-triple-play	0.1	0.1	0.1	0.1	0.1
cube-quadruple-play	0.03	0.03	0.03	0.03	0.03
puzzle-4x5-play	0.01	0.01	0.01	0.01	0.01
puzzle-4x6-play	0.01	0.01	0.01	0.01	0.01
humanoidmaze-giant-navigate	0.01	0.01	0.01	0.01	0.01

C. Additional Results

In this section, we provide the full experimental results across 100 OGBench tasks. Table 7 reports per-task results on 50 standard tasks, where each result is averaged over five independent seeds and accompanied by the corresponding standard deviation. Table 8 presents per-task results on 25 noisy tasks, again averaged over five independent seeds with standard deviations reported. Finally, Table 9 reports per-task results on 25 long-horizon reasoning tasks, where results are averaged over three independent seeds with standard deviations.

Table 7. Full results on standard tasks in OGBench.

	Model-Free			**Model-Based**			**Ablations**				**Ours**
	IQL	ReBRAC	FQL	MOPO	MOBILE	MAC	ReBRAC†	MBTD(λ)	DTD(λ)	GHM	UHM
antmaze-large-navigate-task1	48 ± 9	91 ± 10	80 ± 8	0 ± 0	0 ± 0	17 ± 12	67 ± 41	75 ± 23	98 ± 1	96 ± 1	95 ± 1
antmaze-large-navigate-task2	42 ± 6	88 ± 4	57 ± 10	0 ± 0	0 ± 0	0 ± 0	75 ± 24	68 ± 28	86 ± 4	82 ± 4	80 ± 5
antmaze-large-navigate-task3	72 ± 7	51 ± 18	93 ± 3	0 ± 0	0 ± 0	70 ± 14	78 ± 39	85 ± 14	99 ± 1	93 ± 3	93 ± 3
antmaze-large-navigate-task4	51 ± 9	84 ± 7	80 ± 4	0 ± 0	0 ± 0	0 ± 0	63 ± 35	81 ± 12	90 ± 4	91 ± 2	86 ± 5
antmaze-large-navigate-task5	54 ± 22	90 ± 2	83 ± 4	0 ± 0	0 ± 0	1 ± 1	75 ± 38	45 ± 37	95 ± 2	90 ± 1	91 ± 3
antmaze-giant-navigate-task1	0 ± 0	27 ± 22	4 ± 5	0 ± 0	0 ± 0	0 ± 0	7 ± 11	11 ± 14	27 ± 34	37 ± 14	48 ± 5
antmaze-giant-navigate-task2	1 ± 1	16 ± 17	9 ± 7	0 ± 0	0 ± 0	0 ± 0	26 ± 31	40 ± 17	94 ± 3	1 ± 1	5 ± 6
antmaze-giant-navigate-task3	0 ± 0	34 ± 22	0 ± 1	0 ± 0	0 ± 0	0 ± 0	23 ± 27	14 ± 12	59 ± 17	1 ± 2	7 ± 14
antmaze-giant-navigate-task4	0 ± 0	5 ± 12	14 ± 23	0 ± 0	0 ± 0	0 ± 0	17 ± 22	2 ± 2	30 ± 35	58 ± 27	60 ± 27
antmaze-giant-navigate-task5	19 ± 7	49 ± 22	16 ± 28	0 ± 0	0 ± 0	0 ± 0	79 ± 2	67 ± 17	49 ± 10	67 ± 9	59 ± 14
humanoidmaze-medium-navigate-task1	32 ± 7	16 ± 9	19 ± 12	0 ± 0	0 ± 0	0 ± 1	19 ± 13	34 ± 9	82 ± 17	89 ± 2	95 ± 1
humanoidmaze-medium-navigate-task2	41 ± 9	18 ± 16	94 ± 3	0 ± 0	0 ± 0	2 ± 1	15 ± 15	78 ± 35	95 ± 3	93 ± 1	96 ± 1
humanoidmaze-medium-navigate-task3	25 ± 5	36 ± 13	74 ± 18	0 ± 0	0 ± 0	5 ± 1	37 ± 27	93 ± 2	98 ± 1	93 ± 1	94 ± 2
humanoidmaze-medium-navigate-task4	0 ± 1	15 ± 16	3 ± 4	0 ± 0	0 ± 0	0 ± 0	11 ± 19	19 ± 23	31 ± 22	84 ± 2	92 ± 4
humanoidmaze-medium-navigate-task5	66 ± 4	24 ± 20	97 ± 2	0 ± 0	0 ± 0	2 ± 1	26 ± 22	97 ± 1	98 ± 0	92 ± 2	96 ± 1
humanoidmaze-large-navigate-task1	3 ± 1	2 ± 1	7 ± 6	0 ± 0	0 ± 0	0 ± 0	1 ± 1	25 ± 15	51 ± 22	35 ± 9	69 ± 3
humanoidmaze-large-navigate-task2	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 1	4 ± 5	1 ± 1	3 ± 1
humanoidmaze-large-navigate-task3	7 ± 3	8 ± 4	11 ± 7	0 ± 0	0 ± 0	0 ± 0	3 ± 4	31 ± 10	55 ± 45	40 ± 2	56 ± 17
humanoidmaze-large-navigate-task4	1 ± 0	1 ± 1	2 ± 3	0 ± 0	0 ± 0	0 ± 0	0 ± 0	10 ± 6	8 ± 12	2 ± 1	17 ± 22
humanoidmaze-large-navigate-task5	1 ± 1	2 ± 2	1 ± 3	0 ± 0	0 ± 0	0 ± 0	0 ± 1	12 ± 10	19 ± 25	3 ± 5	17 ± 14
antsoccer-arena-navigate-task1	14 ± 5	0 ± 0	77 ± 4	0 ± 0	0 ± 0	47 ± 6	0 ± 0	53 ± 6	0 ± 0	14 ± 4	17 ± 6
antsoccer-arena-navigate-task2	17 ± 7	0 ± 1	88 ± 3	0 ± 0	0 ± 0	30 ± 10	2 ± 4	66 ± 5	0 ± 0	40 ± 9	50 ± 13
antsoccer-arena-navigate-task3	6 ± 4	0 ± 0	61 ± 6	0 ± 0	0 ± 0	30 ± 2	0 ± 0	56 ± 4	0 ± 0	1 ± 1	8 ± 12
antsoccer-arena-navigate-task4	3 ± 2	0 ± 0	39 ± 6	0 ± 0	0 ± 0	25 ± 1	1 ± 1	31 ± 3	1 ± 1	23 ± 4	25 ± 4
antsoccer-arena-navigate-task5	2 ± 2	0 ± 0	36 ± 9	0 ± 0	0 ± 0	15 ± 8	0 ± 1	27 ± 10	0 ± 0	22 ± 14	28 ± 7
cube-single-play-task1	88 ± 3	89 ± 5	97 ± 2	12 ± 16	85 ± 22	100 ± 0	93 ± 3	91 ± 2	91 ± 7	87 ± 6	95 ± 6
cube-single-play-task2	85 ± 8	92 ± 4	97 ± 2	10 ± 16	80 ± 12	100 ± 0	89 ± 3	93 ± 3	91 ± 4	93 ± 4	94 ± 6
cube-single-play-task3	91 ± 5	93 ± 3	98 ± 2	15 ± 14	83 ± 17	98 ± 3	94 ± 3	96 ± 2	90 ± 6	93 ± 4	93 ± 5
cube-single-play-task4	73 ± 6	92 ± 3	94 ± 3	2 ± 3	72 ± 19	98 ± 3	92 ± 5	92 ± 2	92 ± 5	92 ± 2	88 ± 6
cube-single-play-task5	78 ± 9	87 ± 8	93 ± 3	20 ± 26	87 ± 19	97 ± 7	89 ± 4	86 ± 4	87 ± 6	90 ± 5	89 ± 2
cube-double-play-task1	27 ± 5	45 ± 6	61 ± 9	2 ± 3	7 ± 8	82 ± 15	12 ± 6	12 ± 4	3 ± 2	42 ± 7	44 ± 10
cube-double-play-task2	1 ± 1	7 ± 3	36 ± 6	0 ± 0	0 ± 0	50 ± 12	6 ± 3	3 ± 1	2 ± 1	34 ± 7	34 ± 11
cube-double-play-task3	0 ± 0	4 ± 1	22 ± 5	2 ± 3	0 ± 0	55 ± 10	2 ± 1	1 ± 1	0 ± 1	25 ± 6	26 ± 7
cube-double-play-task4	0 ± 0	1 ± 1	5 ± 2	0 ± 0	0 ± 0	28 ± 8	0 ± 0	0 ± 0	0 ± 0	1 ± 1	1 ± 0
cube-double-play-task5	4 ± 3	4 ± 2	19 ± 10	2 ± 3	0 ± 0	50 ± 9	1 ± 1	1 ± 1	14 ± 3	41 ± 2	43 ± 9
scene-play-task1	94 ± 3	95 ± 2	100 ± 0	30 ± 38	37 ± 16	100 ± 0	82 ± 8	68 ± 14	97 ± 3	89 ± 5	84 ± 4
scene-play-task2	12 ± 3	50 ± 13	76 ± 9	2 ± 3	5 ± 10	100 ± 0	65 ± 5	59 ± 4	96 ± 2	89 ± 4	87 ± 2
scene-play-task3	32 ± 7	55 ± 16	98 ± 1	0 ± 0	0 ± 0	95 ± 10	44 ± 11	26 ± 11	48 ± 12	24 ± 8	23 ± 9
scene-play-task4	0 ± 1	3 ± 3	5 ± 1	0 ± 0	0 ± 0	95 ± 6	9 ± 14	1 ± 1	75 ± 19	3 ± 3	4 ± 6
scene-play-task5	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0	93 ± 8	0 ± 0	0 ± 0	63 ± 9	15 ± 19	19 ± 27
puzzle-3x3-play-task1	33 ± 6	97 ± 4	90 ± 4	100 ± 0	60 ± 47	100 ± 0	96 ± 4	98 ± 1	100 ± 0	93 ± 3	99 ± 1
puzzle-3x3-play-task2	4 ± 3	1 ± 1	16 ± 5	0 ± 0	0 ± 0	0 ± 0	89 ± 8	94 ± 3	100 ± 0	49 ± 2	100 ± 0
puzzle-3x3-play-task3	3 ± 2	3 ± 1	10 ± 3	0 ± 0	0 ± 0	0 ± 0	83 ± 10	85 ± 5	99 ± 1	35 ± 7	99 ± 1
puzzle-3x3-play-task4	2 ± 1	2 ± 1	16 ± 5	0 ± 0	0 ± 0	0 ± 0	89 ± 6	93 ± 3	100 ± 0	35 ± 4	99 ± 1
puzzle-3x3-play-task5	3 ± 2	5 ± 3	16 ± 3	0 ± 0	0 ± 0	0 ± 0	92 ± 1	94 ± 3	99 ± 2	44 ± 7	98 ± 1
puzzle-4x4-play-task1	12 ± 2	26 ± 4	34 ± 8	0 ± 0	0 ± 0	98 ± 3	2 ± 1	9 ± 3	1 ± 1	24 ± 4	19 ± 6
puzzle-4x4-play-task2	7 ± 4	12 ± 4	16 ± 5	0 ± 0	0 ± 0	33 ± 27	1 ± 0	1 ± 1	1 ± 0	11 ± 1	11 ± 5
puzzle-4x4-play-task3	9 ± 3	15 ± 3	18 ± 5	0 ± 0	0 ± 0	100 ± 0	1 ± 0	6 ± 2	1 ± 0	14 ± 2	13 ± 4
puzzle-4x4-play-task4	5 ± 2	10 ± 3	11 ± 3	0 ± 0	0 ± 0	85 ± 14	2 ± 1	3 ± 1	0 ± 1	8 ± 2	8 ± 2
puzzle-4x4-play-task5	4 ± 1	7 ± 3	7 ± 3	0 ± 0	0 ± 0	72 ± 40	1 ± 1	1 ± 1	0 ± 1	8 ± 3	7 ± 2
Average	23	31	44	4	10	40	35 ± 2	45 ± 2	52 ± 1	48 ± 1	55 ± 1

Table 8. Full results on noisy tasks in OGBench.

	ReBRAC†	MBTD(λ)	DTD(λ)	GHM	UHM
antmaze-medium-explore-task1	38 ± 16	98 ± 2	55 ± 25	97 ± 1	95 ± 6
antmaze-medium-explore-task2	97 ± 3	99 ± 1	99 ± 1	96 ± 2	97 ± 1
antmaze-medium-explore-task3	64 ± 16	86 ± 13	75 ± 11	84 ± 12	70 ± 10
antmaze-medium-explore-task4	72 ± 15	98 ± 1	76 ± 14	84 ± 6	89 ± 6
antmaze-medium-explore-task5	98 ± 1	100 ± 0	100 ± 1	94 ± 2	95 ± 6
antmaze-large-explore-task1	3 ± 5	21 ± 13	11 ± 11	22 ± 9	56 ± 16
antmaze-large-explore-task2	15 ± 12	0 ± 0	0 ± 0	0 ± 0	2 ± 3
antmaze-large-explore-task3	40 ± 33	79 ± 12	57 ± 40	48 ± 12	68 ± 10
antmaze-large-explore-task4	5 ± 5	0 ± 0	13 ± 12	0 ± 1	0 ± 0
antmaze-large-explore-task5	1 ± 1	0 ± 0	8 ± 8	0 ± 1	6 ± 5
cube-double-noisy-task1	10 ± 3	9 ± 2	10 ± 7	59 ± 11	52 ± 6
cube-double-noisy-task2	1 ± 1	0 ± 1	1 ± 0	28 ± 10	15 ± 6
cube-double-noisy-task3	0 ± 0	1 ± 1	0 ± 0	11 ± 3	7 ± 1
cube-double-noisy-task4	0 ± 0	0 ± 0	0 ± 0	12 ± 4	9 ± 2
cube-double-noisy-task5	1 ± 1	0 ± 0	0 ± 0	10 ± 3	8 ± 5
scene-noisy-task1	93 ± 3	91 ± 2	26 ± 18	98 ± 1	96 ± 2
scene-noisy-task2	0 ± 0	22 ± 5	0 ± 0	63 ± 8	69 ± 7
scene-noisy-task3	33 ± 32	70 ± 7	5 ± 4	82 ± 6	83 ± 9
scene-noisy-task4	9 ± 8	28 ± 9	0 ± 0	41 ± 9	54 ± 13
scene-noisy-task5	0 ± 0	0 ± 0	0 ± 0	0 ± 0	2 ± 3
puzzle-4x4-noisy-task1	0 ± 0	1 ± 1	0 ± 0	18 ± 3	2 ± 1
puzzle-4x4-noisy-task2	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
puzzle-4x4-noisy-task3	0 ± 0	0 ± 0	0 ± 0	3 ± 2	1 ± 1
puzzle-4x4-noisy-task4	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
puzzle-4x4-noisy-task5	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
Average	23 ± 2	32 ± 1	23 ± 4	38 ± 1	39 ± 1

Table 9. Full results on long horizon tasks in OGBench.

	ReBRAC†	MBTD(λ)	DTD(λ)	GHM	UHM
cube-triple-play-task1	22 ± 6	52 ± 13	3 ± 2	80 ± 14	91 ± 5
cube-triple-play-task2	0 ± 0	2 ± 3	0 ± 0	62 ± 15	94 ± 2
cube-triple-play-task3	0 ± 0	8 ± 6	0 ± 0	61 ± 14	75 ± 6
cube-triple-play-task4	0 ± 0	0 ± 0	1 ± 1	16 ± 7	21 ± 7
cube-triple-play-task5	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
cube-quadruple-play-task1	0 ± 0	0 ± 0	0 ± 0	25 ± 9	34 ± 14
cube-quadruple-play-task2	0 ± 0	0 ± 0	2 ± 3	28 ± 21	27 ± 20
cube-quadruple-play-task3	0 ± 0	0 ± 0	0 ± 0	4 ± 3	13 ± 6
cube-quadruple-play-task4	0 ± 0	0 ± 0	0 ± 0	2 ± 1	4 ± 4
cube-quadruple-play-task5	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
puzzle-4x5-play-task1	36 ± 12	23 ± 14	68 ± 14	83 ± 6	80 ± 2
puzzle-4x5-play-task2	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
puzzle-4x5-play-task3	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
puzzle-4x5-play-task4	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
puzzle-4x5-play-task5	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
puzzle-4x6-play-task1	55 ± 24	5 ± 3	26 ± 28	26 ± 17	55 ± 7
puzzle-4x6-play-task2	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
puzzle-4x6-play-task3	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
puzzle-4x6-play-task4	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
puzzle-4x6-play-task5	0 ± 0	0 ± 0	0 ± 0	0 ± 0	0 ± 0
humanoidmaze-giant-navigate-task1	1 ± 1	0 ± 0	19 ± 8	0 ± 0	3 ± 1
humanoidmaze-giant-navigate-task2	6 ± 5	1 ± 1	58 ± 10	7 ± 1	13 ± 5
humanoidmaze-giant-navigate-task3	0 ± 0	1 ± 1	4 ± 2	0 ± 0	0 ± 0
humanoidmaze-giant-navigate-task4	1 ± 1	2 ± 1	55 ± 7	1 ± 1	1 ± 1
humanoidmaze-giant-navigate-task5	4 ± 4	21 ± 5	94 ± 1	11 ± 2	33 ± 2
Average	5 ± 1	5 ± 1	13 ± 3	16 ± 2	22 ± 1