Long-Horizon Model-Based Offline Reinforcement Learning Without Explicit Conservatism

Anonymous Authors[1]

Abstract

Popular offline reinforcement learning (RL) methods rely on *explicit conservatism*, penalizing out-of-dataset actions or restricting rollout horizons. We question the universality of this principle and revisit a complementary *Bayesian* perspective for test-time adaptation. By modeling a posterior over world models and training a history-dependent agent to maximize expected return, the Bayesian approach directly addresses epistemic uncertainty without explicit conservatism. We first illustrate in a bandit setting that Bayesianism excels on low-quality datasets where conservatism fails. Scaling to realistic tasks, we find that *long-horizon rollouts* are essential to control value overestimation once conservatism is removed. We introduce design choices that enable learning from long-horizon rollouts while mitigating compounding model errors, yielding our algorithm, NEUBAY, grounded in the NEUtral BAYesian principle. On D4RL and NeoRL benchmarks, NEUBAY is competitive with leading conservative algorithms, achieving new state-of-the-art on 7 datasets with rollout horizons of several hundred steps. Finally, we characterize datasets by quality and coverage to identify when NEUBAY is preferable to conservative methods.

1. Introduction

Reinforcement learning (RL) often assumes direct interaction with the environment, which we refer to as online RL (Sutton & Barto, 2018). While successful in simulation, deploying it in real-world settings such as robotics (Dulac-Arnold et al., 2021), recommendation systems (Chen et al., 2024), and language reasoning (Ma et al., 2026) is limited by expensive or risky data collection. A more practical alternative is offline RL (Lange et al., 2012), which learns from pre-collected datasets (e.g., from human demonstrations or prior agents) without further environment access (Levine et al., 2020; Fu et al., 2020; Gulcehre et al., 2020). This decoupling enables safe, scalable training and the potential to outperform the behavior policies that produced the data.

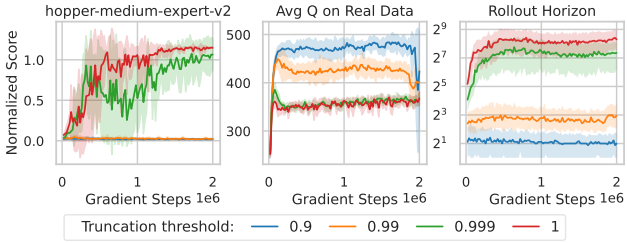

Figure 1. **Adaptive long-horizon rollouts** improve performance (left) with lower estimated Q-value on the offline dataset (middle), **without explicit conservatism**. We vary the rollout truncation threshold $\zeta \in \{0.9, 0.99, 0.999, 1.0\}$ and report horizons (median with 25%–75% band) over 100 training-time rollouts (right). **Full results** are shown in Sec. G.2.

Most offline RL algorithms adopt a conservative principle[1] by penalizing the policy and value function on out-of-dataset state-action pairs (Levine et al., 2020; Prudencio et al., 2023) and using short rollout horizons (Lu et al., 2021). In theory, these algorithms enforce robustness, either strict (Jin et al., 2021; Uehara & Sun, 2022) or soft (Zhang et al., 2024b), over the uncertainty set of possible MDPs consistent with the dataset. The trade-off is clear: conservatism reduces value overestimation and unsafe extrapolation, but can also suppress average-case performance and limit adaptation, since policies are discouraged from exploring potentially high-reward but underrepresented actions.

Bayesian RL optimizes average-case performance under epistemic uncertainty (Duff, 2002). Its application to offline RL was pioneered by Ghosh et al. (2022), who formalized the problem as an *epistemic POMDP*, where partial observability arises from limited coverage. This formulation enables test-time adaptation through history-

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

[1]In this paper, "explicit conservatism" refers to the deliberate injection of pessimism into offline policy learning. Unless otherwise noted, we use "conservatism" as shorthand for this notion. This excludes generic anti-overestimation methods proposed in the online RL literature.

dependent Bayes-optimal policies. In this work, we first revisit and extend this Bayesian principle through the lens of *data quality*. Using a two-armed bandit with skewed data, we show that conservative algorithms, with sufficient uncertainty penalty, are guaranteed to commit to the seen arm regardless of test-time conditions. In contrast, Bayesian algorithms can adaptively explore and commit to the better arm at test time, a clear advantage in low-quality datasets.

However, scaling the Bayesian principle to realistic tasks is challenging, as it requires solving an *approximate* epistemic POMDP. We identify three key challenges: (1) *value overestimation* (Fujimoto et al., 2019), which becomes central once Bayesian RL abandons explicit conservatism and removes penalties for out-of-dataset actions; (2) *compounding error* in world models (Lambert et al., 2022), where inaccuracies grow rapidly with horizon; and (3) training agents with *long-term memory* (Ni et al., 2023) to enable test-time adaptation. Each aligns with a major research area: offline RL, model-based RL, and partially observable RL. This helps explain why prior Bayesian-inspired algorithms often *reintroduce* explicit conservatism through uncertainty penalties and short horizons (Chen et al., 2021c; Jeong et al., 2023) or reduce to model-free RL (Ghosh et al., 2022).

We build a practical algorithm, NEUBAY, to address these challenges. We first show that *long-horizon rollouts* can themselves reduce value overestimation once explicit conservatism is removed. We then introduce design choices that make such long-horizon rollouts feasible. To control compounding errors, we truncate rollouts adaptively using epistemic uncertainty as a threshold (Frauenknecht et al., 2024; Zhan et al., 2021), an alternative use of uncertainty beyond penalties, and apply layer normalization (Ba et al., 2016) within the world model. To stabilize learning from long horizons, we leverage recent advances in online recurrent RL with long-term memory (Morad et al., 2024; Luo et al., 2024a).

We evaluate NEUBAY on the D4RL (Fu et al., 2020) and NeoRL (Qin et al., 2022) benchmarks, covering 33 datasets. Overall, NEUBAY is on par with leading conservative algorithms and outperforms other Bayesian-inspired methods, establishing new state-of-the-art results on 7 datasets. On realistic tasks, NEUBAY performs best on low-quality datasets and on medium-quality datasets with moderate coverage. Our sensitivity study validates a key insight: **adaptive long horizon** is a primary driver of NEUBAY's success. Whereas dominant model-based RL practice favors short horizons, our Bayesian approach uncovers a new role for long horizons: they suppress value overestimation. NEUBAY routinely plans **64-512 steps** (e.g., Fig. 1), while short-horizon variants fail due to severe overestimation. These results position NEUBAY as a practical direction for model-based and offline RL from a Bayesian perspective, and future advances in world modeling can push the limits further.

2. Background on Offline RL

In the standard offline RL setting, a static dataset $\mathcal{D}$ is collected by interacting with an MDP $\mathcal{M}^*$, which we refer to as the *true MDP*. Formally, $\mathcal{M}^* = (\mathcal{S}, \mathcal{A}, \gamma, T, f_{\text{term}}, m^*, \rho)$ where $\mathcal{S}$ and $\mathcal{A}$ are state and action spaces, $\gamma \in (0,1)$ is the discount factor, $T \in \mathbb{N}$ is the maximum episode length, and $f_{\text{term}} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \{0,1\}$ is the terminal function. These components are assumed to be **known** (Puterman, 2014; Yu et al., 2020). The *joint reward-transition function* is $m^* : \mathcal{S} \times \mathcal{A} \to \Delta([-r_{\max}, r_{\max}] \times \mathcal{S})$, consisting of reward function $R^* : \mathcal{S} \times \mathcal{A} \to \Delta([-r_{\max}, r_{\max}])$ (here, $r_{\max}$ is a positive constant) and dynamics $P^* : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$, both **unknown** to the agent and learned in model-based methods. The initial state distribution $\rho \in \Delta(\mathcal{S})$ is also unknown, but we do not model it explicitly since initial states can be directly sampled from $\mathcal{D}$ (Janner et al., 2019).

The static dataset of trajectories[2] $\mathcal{D} = \{\tau^i\}_{i=1}^{\text{num\_traj}}$ is collected by an unknown (possibly) history-dependent behavior policy $\pi_\beta : \mathcal{H}_t \to \Delta(\mathcal{A})$, where $\mathcal{H}_t$ is the space of state-action-reward sequences up to timestep t. Define $h_t = (s_{0:t}, a_{0:t-1}, r_{1:t}) \in \mathcal{H}_t$ for $t \geq 1$ with the convention that $h_0 = s_0$. Each trajectory $\tau = (s_0, a_0, r_1, d_1, s_1, a_1, \dots)$ is generated by: $s_0 \sim \rho, a_t \sim \pi_\beta(h_t), (r_{t+1}, s_{t+1}) \sim m^*(s_t, a_t), d_{t+1} = f_{\text{term}}(s_t, a_t, s_{t+1})$. A trajectory ends either when $d_t = 1$ (termination) or when $t = T$ (truncation). We stress this distinction: *termination* implies absorbing states with zero future rewards, whereas *truncation* preserves continuation and thus allows bootstrapping.

The ideal objective in offline RL is to find a possibly history-dependent policy $\pi : \mathcal{H}_t \to \Delta(\mathcal{A})$ that maximizes the expected discounted return under the true MDP $\mathcal{M}^*$: $\max_\pi J(\pi, m^*) := \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_{t+1}]$, where $s_0 \sim \rho, a_t \sim \pi(h_t), (r_{t+1}, s_{t+1}) \sim m^*(s_t, a_t)$. The defining constraint of offline RL is that the agent cannot interact with m^*, making it intractable to direct optimize. This leads to the following discussion about epistemic uncertainty on m^*.

Empirical model and epistemic uncertainty. From the agent's view, knowledge of m^* is well-defined only on the state-action support of the dataset $\mathcal{D}$: $\text{supp}_{\mathcal{S}\times\mathcal{A}}(\mathcal{D}) := \{(s,a) \mid (s,a,r,s') \in \mathcal{D}\}$. Let $\mathfrak{M}_{\text{in}}$ denote a model class whose domain is restricted to $\text{supp}_{\mathcal{S}\times\mathcal{A}}(\mathcal{D})$. The *empirical model* (Fujimoto et al., 2019) is then obtained by maximum likelihood estimation (MLE): $m_{\mathcal{D}} = \text{argmax}_{m \in \mathfrak{M}_{\text{in}}} \mathbb{E}_{(s,a,r,s') \sim \mathcal{D}}[\log m(r, s' \mid s, a)]$. Thus, $m_{\mathcal{D}}$ is uniquely determined in-support by empirical frequencies, but remains *undefined* for $(s,a) \notin \text{supp}_{\mathcal{S}\times\mathcal{A}}(\mathcal{D})$, giving rise to substantial epistemic

[2] Although offline RL datasets are often stored as transitions, trajectories are available in common benchmarks and used by history-dependent methods (Chen et al., 2021a).

uncertainty (Gal, 2016). A common way to formalize this uncertainty is through an *uncertainty set* $\mathfrak{M}_\mathcal{D}$, the set of plausible models on $\mathcal{S} \times \mathcal{A}$ that agree with $\mathcal{D}$ on $\text{supp}_{\mathcal{S}\times\mathcal{A}}(\mathcal{D})$ (i.e, close to $m_\mathcal{D}$). Offline policy learning then incorporates $\mathfrak{M}_\mathcal{D}$ into optimization, typically via two paradigms: (explicit) *conservatism*, which optimizes against (near) worst-case models in $\mathfrak{M}_\mathcal{D}$, or *Bayesianism*, which leverages a posterior distribution over $\mathfrak{M}_\mathcal{D}$.

Conservative principle: robust MDPs. Conservative RL methods commonly optimize return under the *worst* model of the uncertainty set $\mathfrak{M}_\mathcal{D}$ (Uehara & Sun, 2022):

$$\max_\pi J(\pi; \mathfrak{M}_\mathcal{D}) := \max_\pi \min_{m\in\mathfrak{M}_\mathcal{D}} J(\pi, m). \quad (1)$$

This robust MDP formulation (Wiesemann et al., 2013) covers both model-free (Fujimoto et al., 2019) and model-based algorithms (Rigter et al., 2022; Yu et al., 2020), differing only in the choice of uncertainty set $\mathfrak{M}_\mathcal{D}$ and the degree of robustness (Zhang et al., 2024b). We provide formal connections between conservatism and robustness in Sec. B.1.

Bayesian principle: epistemic POMDPs. Bayesianism instead treats the true model m^* as a random variable (Ghavamzadeh et al., 2015), maintaining a posterior distribution $\mathbb{P}_\mathcal{D}(m)$ over plausible models. Bayesian offline RL (Ghosh et al., 2022; Uehara & Sun, 2022; Fellows et al., 2025) then optimizes the *expected* return under this posterior, known as *ambiguity-neutrality* (Ellsberg, 1961):

$$\max_\pi J(\pi; \mathbb{P}_\mathcal{D}) := \max_\pi \mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}[J(\pi, m)]. \quad (2)$$

The posterior naturally induces an uncertainty set via its support. If $\mathfrak{M}_\mathcal{D} = \text{supp}(\mathbb{P}_\mathcal{D})$, the Bayesian objective (Eq. 2) is less pessimistic than the conservative one (Eq. 1), since for all policies π, $J(\pi; \mathbb{P}_\mathcal{D}) \geq J(\pi; \mathfrak{M}_\mathcal{D})$. Eq. 2 is equivalent to solving a Bayes-adaptive MDP (BAMDP) (Duff, 2002), also known as an epistemic POMDP; see Sec. B.2 for a connection with POMDPs.

Approximate posterior and uncertainty quantifier. To instantiate the Bayesian objective (Eq. 2), we require a tractable approximation $\hat{\mathbb{P}}_\mathcal{D}$ to the posterior over models. Following common practice (Chua et al., 2018; Yu et al., 2020), we approximate $\mathbb{P}_\mathcal{D}$ with a finite ensemble of learned models $\mathbf{m}_{\boldsymbol{\theta}} = \{m_{\theta^n}\}_{n=1}^N$. The ensemble induces the empirical posterior $\hat{\mathbb{P}}_\mathcal{D}(m) = \frac{1}{N}\sum_{n=1}^N \mathbb{1}(m = m_{\theta^n})$ that captures modes of true posterior (Fort et al., 2019; Wilson & Izmailov, 2020). Given a state-action pair (s, a), we adopt the disagreement among the *mean* predictions (Lakshminarayanan et al., 2017) as uncertainty quantifier, $U_{\boldsymbol{\theta}}(s, a) = \text{std}(\{\mu_{\theta^n}(s, a)\}_{n=1}^N)$, which is shown to be empirically robust under dataset shift (Ovadia et al., 2019). We defer architectural and training details of the ensemble to Sec. F.3.

3. Illustrative Example for Bayesianism

The bandit dataset. We consider a sequential two-armed bandit with $\mathcal{A} = \{0, 1\}$. Each arm $a \in \mathcal{A}$ yields a Bernoulli reward $r \sim R^*(a) = \mathcal{B}(p_a^*)$, so the true MDP m^* is specified by reward parameters $p^* \in [0, 1] \times [0, 1]$. We fix $p_0^* = 0.5$ in our setup. The optimal policy π^* in m^* is deterministic and memoryless, always choosing $\text{argmax}_{a\in\mathcal{A}} p_a^*$. To highlight out-of-support challenges, we construct a dataset that only covers arm 0. Specifically, the dataset $\mathcal{D} = \{(a^i_{0:T-1}, r^i_{1:T})\}_i$ is collected by a deterministic behavior policy $\pi_\beta(a) = \mathbb{1}(a = 0)$. Thus, for each $t < T = 100$, $a_t = 0$, $r_{t+1} \sim \mathcal{B}(p^*_{a_t})$, and $\mathcal{D}$ contains no data on $a = 1$.

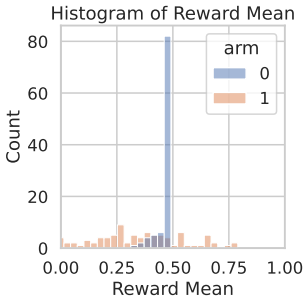

Figure 3. Histogram of estimated reward means p_0, p_1 across ensemble members.

This skewed dataset $\mathcal{D}$ leaves the true reward parameter p_1^* for arm 1 *completely* unobserved, inducing substantial epistemic uncertainty on p_1^*. We approximate the posterior by fitting an ensemble of reward models $\mathbf{m}_{\boldsymbol{\theta}}$ using Gaussian outputs. As shown in Fig. 3, after training on $\mathcal{D}$, the ensemble predictions concentrate around the observed arm 0 but sharply disagree on the unobserved arm 1, with estimated uncertainty about $10\times$ larger than on arm 0. Next, we test the Bayesian method from Eq. 2, where planning begins at $t = 0$ and truncates at $t = T$. Dynamics are not modeled due to the bandit structure, and there is no concern about compounding error.

Uncertainty penalty hurts generalization, but Bayesian agents adapt to worse *and* better test cases. We study explicit conservatism by penalizing the reward with an uncertainty-based term $\lambda U_{\boldsymbol{\theta}}(a)$. Such a design is common in empirical algorithms (Yu et al., 2020), and can be viewed as soft robustness (Zhang et al., 2024b) (see Sec. B.1). Since the estimated uncertainty on arm 1 is much larger than arm 0 (i.e, $U_{\boldsymbol{\theta}}(1) \gg U_{\boldsymbol{\theta}}(0)$), a sufficiently large coefficient λ forces the resulting algorithm to always select arm 0, regardless of the test-time environment. In contrast, the Bayesian agent ($\lambda = 0$) trains on a posterior $\mathbb{P}_\mathcal{D}(p_1)$ that spans both worse and better values relative to p_0^* (see Fig. 3). It adopts a history-dependent policy that explores the unseen arm at test time before committing, similarly to meta-RL (Zintgraf et al., 2020). Thus, as shown in Fig. 2, the Bayesian agent's return is slightly worse than conservatism when $p_1^* < p_0^* = 0.5$, because it briefly explores arm 1. When $p_1^* > 0.5$, Bayesianism is a clear win, as it identifies and exploits the better unseen action, whereas

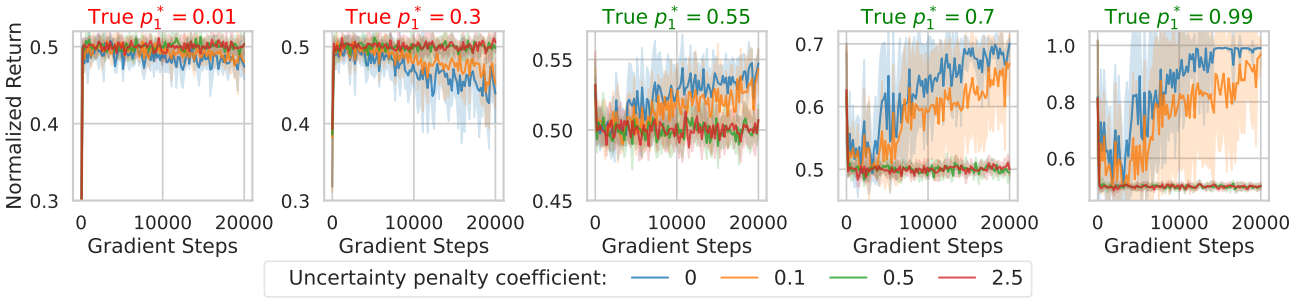

Figure 2. Average return (normalized by T) on **test-time bandits** with different p_1^*. Since the observed arm has $p_0^* = 0.5$, cases with $p_1^* < 0.5$ are *worse* and those with $p_1^* > 0.5$ are *better*. Zero penalty: Bayesian agent; nonzero penalty: conservative agent.

heavy conservatism remains stuck on arm 0. This leads to our main insight: *Bayesianism excels on **low-quality** datasets or when optimal actions are scarce by leveraging test-time adaptation, while remaining competitive on **high-quality** datasets at the cost of exploration.*

We theoretically confirm this empirical insight in Sec. D by comparing robust-optimal and Bayes-optimal policies under the true MDP. For skewed datasets such as the bandits above, we show that Bayes-optimal policies outperform conservative ones by a provable margin. This gap is characterized by two data-quality measures: the Bayesian- and robust-coverage ratios. The robust-coverage ratio can grow arbitrarily large, while posterior averaging bounds the Bayesian-coverage ratio, explaining the performance difference.

4. A Practical Bayesian-Principled Algorithm

The Bayesian agent is conceptually simple and, as seen in the bandit example, clearly outperforms conservative agents on low-quality data and enables test-time adaptation. Extending this principle from bandits to realistic MDPs, however, is challenging: the full-horizon rollouts implied by the Bayesian objective (Eq. 2) can suffer from severe compounding model errors (Luo et al., 2024b; Lin et al., 2025). While this may suggest restricting to short-horizon rollouts, we first show that **long-horizon rollouts** are in fact necessary for Bayesian agents, as longer horizons reduce value overestimation, a central challenge once explicit conservatism is removed (Sec. 4.1). We then introduce a set of design choices for both performing and learning from long-horizon rollouts (Sec. 4.2–Sec. 4.4), yielding our practical algorithm, NEUBAY (Sec. 4.5).

4.1. Why Do We Need Long-Horizon Rollouts?

Ideally, under a correct posterior, the Bayesian principle calls for full-horizon rollouts. In practice, however, compounding errors make full-horizon rollouts unreliable. To mitigate this issue, prior work in offline RL typically adopts a *combination* of short-horizon rollouts and conservatism: short horizons limit compounding errors, while explicit conservatism (e.g., uncertainty penalties) significantly reduces value overestimation caused by the extrapolation error (Fujimoto et al., 2019; Kumar et al., 2019).

In contrast, our focus is on Bayesian agents that aim to adapt beyond the dataset, as illustrated in Sec. 3. This requires abandoning conservatism, but doing so removes an explicit mechanism for controlling value overestimation. We show that, perhaps surprisingly, *long-horizon rollouts* can themselves serve this role by actively **reducing overestimation** compared to short-horizon rollouts.

To illustrate, starting from a real history $h_t \in \mathcal{D}$, we sample a model m_θ from the ensemble $\mathbf{m_\theta}$ and generate a trajectory of length H, where $\hat{a}_{t+j} = \pi(\hat{h}_{t+j})$ comes from a deterministic policy, $(\hat{r}_{t+j+1}, \hat{s}_{t+j+1}) \sim m_\theta(\hat{s}_{t+j}, \hat{a}_{t+j})$, and $\hat{h}_t = h_t$. Applying one-step Bellman backups on the Bayesian value function *along this rollout*, i.e., $Q^{\text{Bayes}}(\hat{h}_{t+j}, \hat{a}_{t+j}) \leftarrow \hat{r}_{t+j+1} + \gamma Q^{\text{Bayes}}(\hat{h}_{t+j+1}, \pi(\hat{h}_{t+j+1}))$, $0 \le j < H$, we obtain an H-step TD target on $Q^{\text{Bayes}}(h_t, \hat{a}_t)$:

$$\sum_{j=0}^{H-1} \gamma^j \underbrace{\hat{r}_{t+j+1}}_{\text{lower bias}} + \underbrace{\gamma^H}_{\text{discounted}} \underbrace{Q^{\text{Bayes}}(\hat{h}_{t+H}, \pi(\hat{h}_{t+H}))}_{\text{higher bias}}. \quad (3)$$

This decomposition highlights the bias trade-off: imagined rewards can be low-bias if the model generalizes, while the bootstrapped term – more susceptible to overestimation (Kumar et al., 2019; Sims et al., 2024) – is exponentially discounted with H. We formalize this effect in Sec. C, extending the analysis of Sims et al. (2024). Importantly, this role of long-horizon rollouts in mitigating overestimation is distinct from their use for data augmentation or exploration (Young et al., 2023). Rather than relying on conservatism, long horizons absorb overestimation risk by shifting estimation from bootstrapping toward model-generated rewards. The cost of longer horizons is increased compounding error, which we address in Sec. 4.2 and Sec. 4.3.

4.2. How Do We Perform Long-Horizon Rollouts?

Where to start planning? From the Bayesian objective (Eq. 2), it is natural to initiate planning rollouts by sampling initial states $s_0 \sim \rho_{\mathcal{D}}$, where $\rho_{\mathcal{D}}$ is the empirical initial-state distribution. But this underrepresents states appearing *later* in real trajectories, as compounding errors make them harder to reach. Following MBPO-style branching mechanism (Janner et al., 2019), we sample starting states $s_t \sim \mathcal{D}$ from any timestep t to maintain coverage.

Since the environment appears as an epistemic POMDP to the *agent*, we also provide the corresponding history $h_t = (s_{0:t}, a_{0:t-1}, r_{1:t}) \in \mathcal{D}$.

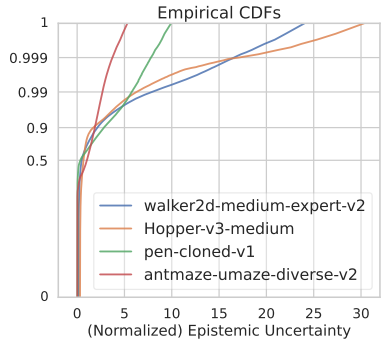

Figure 4. **Empirical CDFs** of epistemic uncertainty $U_{\boldsymbol{\theta}}$ over in-dataset (s, a) (logit-scaled y-axis). Uncertainties are normalized by the dataset mean, so 1 is the average value.

How to truncate long-horizon rollouts? Since model errors depend on specific (s, a) pairs, a key question is not simply *when* to truncate rollouts, but *where*. A natural criterion is the model's uncertainty estimate $U_{\boldsymbol{\theta}}(s, a)$, which correlates with prediction error. In practice, epistemic uncertainty is highly *non-uniform* even within $\mathcal{D}$: frequently visited (s, a) pairs yield low uncertainty, while rarely seen ones result in high uncertainty, reflecting uncertainty about the true model m^* rather than $m_{\mathcal{D}}$.[3] As shown in Fig. 4, most empirical CDFs are long-tailed, with dataset-dependent skewness reflecting the underlying visitation distribution.

Prior work has also used uncertainty threshold to limit model error, together with short horizon caps (Pan et al., 2020; Zhan et al., 2021; Zhang et al., 2023; Frauenknecht et al., 2024). At a high level, these methods and ours share the same principle: using the longest reliable rollout before model error becomes harmful. Our setting differs in how this reliable horizon is used. Once explicit conservatism is removed, short horizons can cause severe overestimation due to excessive reliance on bootstrapping. We therefore remove any fixed cap and use uncertainty alone to determine truncation adaptively, allowing rollouts to remain long wherever the model is reliable.

Uncertainty threshold $\mathcal{U}(\zeta)$ for rollout truncation (design choice 1). Let $\zeta \in [0, 1]$, we define

$$\mathcal{U}(\zeta) := F_Y^{-1}(\zeta), \quad Y := U_{\boldsymbol{\theta}}(s, a),\ (s, a) \sim \mathcal{D},$$

where F_Y^{-1} is the quantile function of Y. Quantile-based thresholds adapt naturally to different datasets and uncertainty scales. Setting $\zeta = 1.0$ yields the maximum in-dataset uncertainty and encourages the longest possible rollouts; thresholds beyond this are avoided as they correspond to out-of-distribution uncertainty.

[3] Standard concentration bounds (Kumar et al., 2020, Section D.3) imply that, with probability $1-\delta$, for $(s, a) \in \mathrm{supp}_{\mathcal{S}\times\mathcal{A}}(\mathcal{D})$, $\mathrm{TV}(m_{\mathcal{D}}(s, a), m^*(s, a)) \leq c_{m^*,\delta}/\sqrt{n_{\mathcal{D}}(s, a)}$, where $n_{\mathcal{D}}(s, a)$ is the visitation count in $\mathcal{D}$ and $c_{m^*,\delta}$ is a constant.

4.3. World Architecture for Long-Horizon Rollouts

To support long rollouts, we introduce simple and effective architectural choices for the world model ensemble that mitigate compounding errors and improve posterior fidelity.

Larger ensemble size N (design choice 2). While deep ensembles approximate Bayesian posteriors, their fidelity relies on member diversity. Under long-horizon rollouts, compounding errors amplify posterior inaccuracies, rendering small ensembles (e.g., $N = 5$ in MBPO (Janner et al., 2019)) inadequate. In our main experiments, we therefore use a larger ensemble size ($N = 100$), which remains computationally feasible due to parallelization.

Layer normalization in the world model (design choice 3). We further find that applying layer normalization (LN) (Ba et al., 2016) within the world model significantly mitigates compounding errors for long-horizon rollouts. This mirrors its role in reducing extrapolation error in model-free RL (Ball et al., 2023). Concretely, we structure the world model as a delta predictor and apply LN to its hidden features. The model predicts the next state as $\mathbb{E}[\hat{s}'] = s + \mathbf{W}^\top \mathrm{ReLU}(\mathrm{LN}(\psi(s, a))$, with features $\psi(s, a) \in \mathbb{R}^k$ and output weights $\mathbf{W} \in \mathbb{R}^{k\times|\mathcal{S}|}$. For LN without affine parameters, under ℓ_2 norm:

$$\begin{aligned} \|\mathbb{E}[\hat{s}'] - s\| &\leq \|\mathbf{W}\|\|\mathrm{ReLU}(\mathrm{LN}(\psi(s, a))\| \\ &\leq \|\mathbf{W}\|\|\mathrm{LN}(\psi(s, a))\| = \sqrt{k}\|\mathbf{W}\|, \end{aligned} \tag{4}$$

using that fact that $\|\mathrm{LN}(x)\| = \sqrt{k}$ for any $x \in \mathbb{R}^k$. Applying the triangle inequality over an H-step imagined trajectory yields a linear compounding bound, $\|\mathbb{E}[\hat{s}_H] - s_0\| \leq H\sqrt{k}\|\mathbf{W}\|$. Therefore, by controlling $\|\mathbb{E}[\hat{s}_H]\|$, we can upper bound the compounding error: $\|\mathbb{E}[\hat{s}_H] - s_H\| \leq \|\mathbb{E}[\hat{s}_H]\| + \|s_H\|$.

4.4. Stable Recurrent RL Using Long-Horizon Rollouts

Lastly, we describe how we train policies from long-horizon rollouts. Following MBPO (Janner et al., 2019), we apply model-free RL to a mixture of real and model-generated data. Since real data is off-policy and $\hat{\mathbb{P}}_{\mathcal{D}}$ induces a POMDP, we use recurrent off-policy RL with a recurrent actor $\pi_\nu(a_t \mid h_t)$ and critic $Q_\omega(h_t, a_t)$, each equipped with a separate RNN encoder ($\nu_\phi(h_t)$ and $\omega_\phi(h_t)$) for stability (Ni et al., 2022). Long rollouts and test-time adaptation require memory spanning entire episodes (up to 1000 steps in our tasks), beyond the capacity of *standard* RNNs (Ni et al., 2023). We therefore extend memoroid (Morad et al., 2024) to actor-critic architectures using linear recurrent units (LRUs) (Orvieto et al., 2023), enabling long-term memory with parallel optimization. Since memoroid was originally designed for online POMDPs, this mismatch motivates additional design choices below.

Balancing real and imagined data with $\kappa \in (0, 1)$ (design choice 4). Following MBPO, we introduce a mixing ratio

κ to weight real versus imagined transitions in the RL loss (see Sec. E). Prior work (Zhang et al., 2023) shows that when real data is of higher quality, a larger κ is preferable.

Small context encoder learning rate η_ϕ (design choice 5). Borrowing insights from online recurrent RL (Luo et al., 2024a), we use a much smaller learning rate for recurrent encoders, since history representations can diverge exponentially with history length even under tiny parameter changes.

Algorithm 1 NEUBAY: Full training loop

Require: Offline dataset $\mathcal{D}$, Online buffer $\mathcal{B} \leftarrow \emptyset$. World ensemble: $\mathbf{m_\theta}$, Recurrent actor: π_ν, Recurrent critic: Q_ω.
Train $\mathbf{m_\theta}$ on $\mathcal{D}$ until convergence
while gradient steps $\leq$ max gradient steps **do**
  for $k = 1$ to K rollouts **do** ▷ Parallelized in practice
    Sample $m_\theta \sim \mathbf{m_\theta}$, fixed for the rollout
    Sample initial history $h_t \sim \mathcal{D}$
    Append ROLLOUT(h_t, m_θ, π_ν) to $\mathcal{B}$
  end for
  for $g = 1$ to G gradient steps **do**
    Optimize RL loss $L(Q_\omega, \pi_\nu; \tau, \kappa)$ with $\tau \sim \mathcal{B}$
  end for
end while

Algorithm 2 NEUBAY: Rollout function

Require: Terminal function f_{term}, Max episode length T, Uncertainty threshold $\mathcal{U}(\zeta)$, ~~Rollout horizon H~~
function ROLLOUT(h_t, m_θ, π_ν)
  Set $\hat{h}_t = h_t$ and done to False
  while done is False **do**
    $\hat{a}_t \sim \pi_\nu(\hat{h}_t), (\hat{r}_{t+1}, \hat{s}_{t+1}) \sim m_\theta(\hat{s}_t, \hat{a}_t)$
    $\hat{d}_{t+1} = f_{\text{term}}(\hat{s}_t, \hat{a}_t, \hat{s}_{t+1})$
    $\text{trunc} = (U_{\boldsymbol{\theta}}(\hat{s}_t, \hat{a}_t) > \mathcal{U}(\zeta))$
    $\text{done} = \hat{d}_{t+1} \vee \text{trunc} \vee (t+1 \geq T)$
    $\hat{h}_{t+1} = (\hat{h}_t, \hat{a}_t, \hat{r}_{t+1}, \hat{s}_{t+1}), t \leftarrow t+1$
  end while
  return Trajectory τ (i.e., h_t plus imagined rollout)
end function

4.5. Overall Algorithm

We now present the full method in Algo. 1, integrating all design choices with the core rollout subroutine (Algo. 2). NEUBAY remains conceptually simple, using single-step models and no uncertainty penalty, while addressing the challenges of Bayesian offline RL. We highlight the lines where NEUBAY differs from prior work. In particular, to respect the Bayesian objective, we fix a single model per rollout, rather than randomizing the model index at each step as in MBPO-style methods. For completeness, Sec. E provides details on RL loss and stopping criteria, although they are not needed to follow the experiments below.

5. Experiments

Benchmarks. We evaluate on two widely used offline RL suites for continuous control: D4RL (Fu et al., 2020) and NeoRL (Qin et al., 2022). From **D4RL**, we use the **locomotion** benchmark (**12** datasets) spanning three MuJoCo environments: halfcheetah (hc), hopper (hp), and walker2d (wk). Each one provides four data regimes: random, medium-replay, medium, and medium-expert. From **NeoRL**, we use its **locomotion** benchmark (**9** datasets) built on the same environments, but with data collected from more deterministic policies, resulting in narrower coverage and different data regimes: Low, Medium, High. We also include the **Adroit** benchmark (**6** datasets) from D4RL, where a 28-DoF robotic arm manipulates objects to solve tasks (pen, door, hammer) with data collected from human demonstrations (human) and behavior cloning (cloned). The difficulty of Adroit lies in its high-dimensional control, small data size, and low-data quality. Lastly, we perform experiments on **AntMaze** (**6** datasets), where an 8-DoF ant robot must reach a goal position in a maze. Tasks span maze layouts of different sizes (umaze, medium, large) and start distributions (play, diverse), and are challenging for model-based RL due to navigation under sparse rewards. See more details in Sec. F.2.

Baselines. We compare against offline RL baselines grouped into three categories, including **conservative model-free RL** (behavior cloning (BC), CQL (Kumar et al., 2020), IQL (Kostrikov et al., 2022), EDAC (An et al., 2021), ReBRAC (Tarasov et al., 2023a)), **conservative model-based RL** (MOPO (Yu et al., 2020), COMBO (Yu et al., 2021), RAMBO (Rigter et al., 2022), ARMOR (Bhardwaj et al., 2023), MOBILE (Sun et al., 2023), LEQ (Park & Lee, 2025), MoMo (Srinivasan & Knottenbelt, 2024), ADMPO (Lin et al., 2025), SUMO (Qiao et al., 2025), VIPO (Chen et al., 2025), ScorePen (Liu et al., 2025), ROMI (Qiao et al., 2026)), and **Bayesian-inspired RL** (APE-V (Ghosh et al., 2022), MAPLE (Chen et al., 2021c), CBOP (Jeong et al., 2023), MoDAP (Choi et al., 2024)).

Algorithm setup. In our main experiments (Sec. 5.1), we adopt the following **default** design choices: ensemble size $N = 100$, layer normalization in the world models, uncertainty threshold $\mathcal{U}(\zeta)$ of $\zeta = 1.0$. The learning rates of the actor and critic MLP heads are fixed to 1×10^{-4}, while their RNN encoders share a tied learning rate η_ϕ swept over $[3 \times 10^{-7}, 1 \times 10^{-4}]$, the exact range being benchmark-dependent. Similarly, the real data ratio κ is swept within $[0.05, 0.95]$, also depending on the benchmark. Note that this yields *two* per-dataset hyperparameters, consistent with common practice in conservative model-based RL (Yu et al., 2020; Rigter et al., 2022; Sun et al., 2023).

Following common practice in offline RL benchmarking, including the baselines we compare against, we tune these hyperparameters based on final policy performance in the true environment. We report the best results for each dataset. Implementation details are provided in Sec. F. In Sec. 5.2, we conduct sensitivity and ablation studies on design choices.

5.1. Benchmarking Results

We present results for D4RL locomotion in Tab. 1, NeoRL locomotion in Tab. 10, D4RL Adroit in Tab. 11, and

Table 1. Comparison of offline RL methods on the **D4RL locomotion** benchmark. We report mean normalized scores for all baselines, with $\pm$std for competitive baselines. The **best mean score** is bolded, and marked methods are statistically similar under a t-test. Our results use 6 seeds, each evaluated at the final step with 20 episodes.

	Model-free		Conservative model-based									Bayesian-inspired				Ours
Dataset	CQL	EDAC	COMBO	RAMBO	MOPO	LEQ	MOBILE	MoMo	ADMPO	SUMO	VIPO	APE-V	MAPLE	MoDAP	CBOP	NEUBAY
hc-random	31.3	28.4	38.8	40.0	38.5	30.8±3.3	39.3±3.0	39.6±3.7	**45.4**±2.8	34.9±2.1	42.5±0.2	29.9	41.5	36.5±1.8	32.8±0.4	37.0±3.3
hp-random	5.3	25.3	17.9	21.6	31.7	32.4±0.3	31.9±0.6	18.3±2.8	32.7±0.2	30.8±0.9	**33.4**±1.9	31.3	10.7	8.9±1.1	31.4±0.0	24.5±28.5
wk-random	5.4	16.6	7.0	11.5	7.4	21.5±0.1	17.9±6.6	26.8±3.3	22.2±0.2	27.9±2.0	20.0±0.1	15.5	22.1	23.1±1.6	17.8±0.4	**34.1**±6.8
hc-med-rep	45.3	61.3	55.1	**77.6**	72.1	65.5±1.1	71.7±1.2	72.9±1.8	67.6±3.4	76.2±1.3	77.2±0.4	64.6	69.5	67.3±3.4	66.4±0.3	72.1±2.4
hp-med-rep	86.3	101.0	89.5	92.8	103.5	103.9±1.3	103.9±1.0	104.0±1.8	104.4±0.4	109.9±1.4	109.6±0.9	98.5	85.0	94.2±4.8	104.3±0.4	**110.6**±0.7
wk-med-rep	76.8	87.1	56.0	86.9	85.6	98.7±6.0	89.9±1.5	90.4±7.7	95.6±2.1	78.2±1.5	98.4±0.3	82.9	75.4	88.4±4.2	92.7±0.9	**99.3**±19.3
hc-medium	46.9	65.9	54.2	68.9	73.0	71.7±4.4	74.6±1.2	77.1±0.9	72.2±0.6	**84.3**±2.4	80.0±0.4	69.1	48.5	77.3±1.1	74.3±0.2	78.6±1.6
hp-medium	61.9	101.6	97.2	96.6	62.8	103.4±0.3	106.6±0.6	**110.8**±2.3	107.4±0.6	104.8±2.1	107.7±1.0	—	44.1	106.6±1.9	102.6±0.1	54.2±7.2
wk-medium	79.5	92.5	81.9	85.0	84.1	74.9±26.9	87.7±1.1	95.0±1.4	93.2±1.1	94.1±2.5	93.1±1.8	90.3	81.3	81.1±6.5	95.5±0.4	**106.4**±23.0
hc-med-exp	95.0	106.3	90.0	93.7	90.8	102.8±0.4	108.2±2.5	107.9±1.9	103.7±0.2	106.6±2.4	**110.0**±0.4	101.4	55.4	103.4±4.3	105.4±1.6	109.5±8.7
hp-med-exp	96.9	110.7	111.1	83.3	81.6	109.4±1.8	112.6±0.2	109.1±0.4	112.7±0.3	107.8±0.7	113.2±0.1	105.7	95.3	94.5±7.8	111.6±0.2	**114.8**±0.5
wk-med-exp	109.1	114.7	103.3	68.3	112.9	108.2±1.3	115.2±0.7	118.4±0.9	114.9±0.3	**122.8**±0.4	117.7±1.0	110.0	107.0	112.2±2.8	117.2±0.5	120.7±1.3
AVG	61.6	76.0	66.8	68.9	70.3	76.9	80.0	80.9	81.0	81.5	**83.6**	—	61.3	74.5	79.3	80.1

D4RL AntMaze in Tab. 12. Overall, NEUBAY is competitive with leading conservative methods. The average normalized scores are 80.1 vs. 83.6 for the best baseline (VIPO) in D4RL locomotion, 64.7 vs. 73.3 for the best baseline (VIPO) in NeoRL locomotion, 21.1 vs. 28.1 for the best model-based baseline (MoMo) in D4RL Adroit, and 28.8 vs. 64.9 for the best model-based baseline (LEQ) in D4RL AntMaze. NEUBAY establishes **new state-of-the-art** (SOTA) mean performance on 7 of the 33 datasets, with 3 gains statistically significant. In terms of magnitude, it advances walker-random-v2 (27.9 $\rightarrow$ 34.1), walker-medium-v2 (95.5 $\rightarrow$ 106.4), pen-cloned-v1 (74.1 $\rightarrow$ 91.3), and hammer-cloned-v1 (5.0 $\rightarrow$ 14.4). It also surpasses prior Bayesian-inspired methods on most tasks, despite their reliance on explicit conservatism.[4] To ensure fair comparison with conservative approaches, we also evaluate a design-matched conservative variant of NEUBAY by incorporating uncertainty penalties while keeping all other design choices identical (see Sec. 5.2).

When is NEUBAY better than conservative methods? We group the 33 datasets into 3 categories:

- **Low-quality datasets (11 tasks):** 3 D4RL random, 3 NeoRL Low, 2 Adroit door, 2 Adroit hammer, and AntMaze umaze-diverse. Consistent with the bandit insight (Sec. 3), NEUBAY ranks among the best methods in 5 of 11 tasks and achieves reasonable performance in 10 of 11.
- **Medium-quality, moderate coverage (7 tasks):** 3 D4RL medium-replay, 3 D4RL medium-expert[5], and Adroit pen-cloned. NEUBAY ranks among the best methods in 5 of 7 tasks, without any failures.
- **Medium-quality, narrow coverage (15 tasks):** 3 D4RL medium, 3 NeoRL Medium, 3 NeoRL High, Adroit pen-human, 5 AntMaze tasks. NEUBAY is weaker, ranking among the best methods in only 3 of 15 tasks. In AntMaze, narrow coverage and contact-rich dynamics make world modeling difficult, while sparse rewards hinder exploration (see Sec. G.5 for analysis). Adding uncertainty penalties in NEUBAY further degrades performance (Tab. 2), suggesting that AntMaze remains challenging for many model-based RL methods.

[4] An exception is MoDAP, which avoids explicit conservatism by fine-tuning model during policy learning.

[5] D4RL expert and NeoRL High are treated as medium-quality, as their behavior policies are not fully optimal.

5.2. What Matters in NEUBAY?

To understand which components are critical to the success of NEUBAY, we conduct sensitivity studies on the design choices introduced in Sec. 4 and ablation studies on the uncertainty penalty and Markov agent (shown in Sec. G.4), using the best hyperparameters identified in Sec. 5.1.

Adaptive long-horizon rollout (Sec. 4.2) is the *decisive* factor for success. We vary the uncertainty quantile $\zeta \in \{0.9, 0.99, 0.999, 1.0\}$ and find that the most aggressive choice, $\zeta = 1.0$, yields the best performance on almost all datasets. As shown in Fig. 1 (with full results in Sec. G.2 and a summary in Tab. 2), $\zeta = 1.0$ yields **64-512 steps** for 75th-percentile horizon and **256-1000 steps** for maximum horizon, in 21 out of 23 tasks with $T = 1000$. These horizons are far longer than typically used in model-based RL, contradicting the conventional view that long rollouts are harmful when horizons are fixed (Janner et al., 2019). With *adaptive* horizons, compounding errors remain controlled, allowing rollouts to extend deep into the episode.

In contrast, performance often collapses to *near zero* (i.e., scores ≤ 5) with smaller thresholds such as $\zeta = 0.9$ (16 datasets) or 0.99 (6 datasets). The consistent failure mode behind these cases is **severe value overestimation**, where estimated Q-values ($\mathbb{E}_{(h_t,a_t)\sim\mathcal{D}}[Q_\omega(h_t, a_t)]$) are

Table 2. **Sensitivity and ablation results averaged across benchmarks.** The highlighted setting (N=100, λ=0.0, ζ=1.0, using the entire history as agent input) is the main result. Ablations vary one hyperparameter at a time, except for the Markov agent, where we sweep the real data ratio κ for a fair comparison. Shading shows degradation level: light (3–10), medium (10–30), dark (>30). Full per-dataset results appear in Tab. 13.

	Ensemble size N			Unc. penalty coef. λ					Truncation threshold ζ				Agent input	
Benchmark	100	20	5	0.0	0.04	0.2	1.0	5.0	1.0	0.999	0.99	0.9	Hist.	Mark.
D4RL Locomotion (12 tasks)	80.1	71.6	65.9	80.1	79.8	80.4	73.1	57.4	80.1	69.6	45.1	22.5	80.1	75.8
NeoRL Locomotion (9 tasks)	64.7	60.3	59.8	64.7	61.5	59.8	58.3	42.5	64.7	58.8	36.3	16.7	64.7	66.8
D4RL Adroit (3 non-zero tasks)	42.2	30.0	32.6	42.2	33.7	34.2	38.1	34.1	42.2	17.8	16.0	1.9	42.2	36.4
D4RL AntMaze (4 non-zero tasks)	43.2	33.2	28.8	43.2	5.0	3.2	4.2	12.0	43.2	35.8	38.4	32.7	43.2	1.3

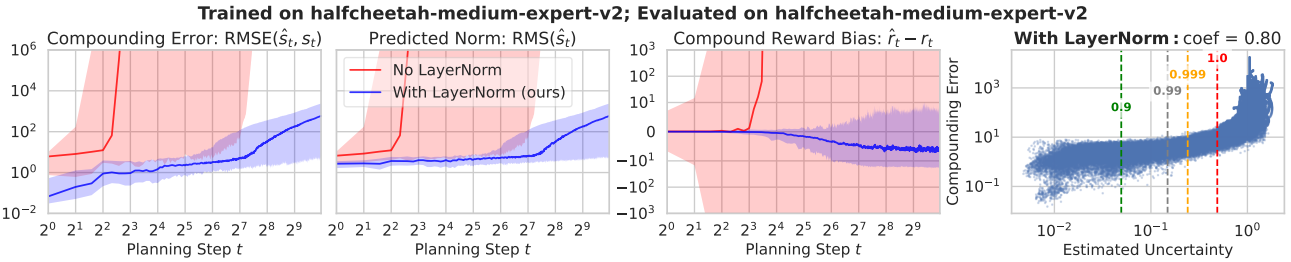

Figure 5. **Effect of LayerNorm in world models** trained and evaluated on halfcheetah-medium-expert-v2. We collect 200 rollouts and truncate only on `float32` overflow, without using an uncertainty threshold. For each metric, we plot the **median** (solid line) together with the **5-95% percentile band** across rollouts. The rightmost scatter plot show the Spearman's rank coefficient in the with-LayerNorm setting; vertical lines mark uncertainty thresholds $\zeta \in \{0.9, 0.99, 0.999, 1.0\}$. **Full results** and plotting setup are shown in Sec. G.3.

much higher (2nd column in Fig. 1) compared to the actual performances. These results support our analysis in Sec. 4.1: without explicit conservatism, Bayesian RL mitigates overestimation by relying more on imagined rewards over long horizons, which are lower-biased (reward panel in Fig. 5). Adaptive horizons ensure that the model is trusted only within its in-distribution confidence region, enabling effective long-horizon rollouts.

LayerNorm in the world model (Sec. 4.3) and uncertainty truncation (Sec. 4.2) jointly enables long-horizon rollouts. Fig. 5 (see Sec. G.3 for full results) shows that without LN, predicted state norms diverge quickly (around 10 steps here), driving exploding compounding errors. With LN, state norms remain bounded, which suppresses state error growth and stabilizes reward predictions. This matches our intuition based on Eq. 4: by normalizing features at each step, LN constrains prediction magnitudes and thus compounding error. Moreover, the rightmost scatter plot shows what uncertainty-based truncation *would* accomplish: although we display full rollouts to reveal their divergence, applying thresholds would cut them off before entering high-error regions. Thus, LayerNorm prevents error explosion, while uncertainty cutoff provides a complementary safeguard.

Larger ensemble size (Sec. 4.3) improves performance. As summarized in Tab. 2 and detailed in Tab. 13, reducing the ensemble size N to 20 or 5 degrades performance, though often moderately. This indicates that $N = 100$ is close to the practical limit of what ensembling can offer for these tasks, while smaller ensembles remain viable.

Context encoder learning rate and real data ratio (Sec. 4.4) have to be tuned per dataset. The best values of η_ϕ and κ are reported in Tab. 4–Tab. 6, with selective learning curves shown in Fig. 13–Fig. 14. We find the optimal encoder learning rates are generally smaller than those used in online POMDPs (Luo et al., 2024a); for example, 1×10^{-6} and 3×10^{-7} yield the best results on 7 datasets. Intuitively, in Bayesian offline RL, smaller learning rates help curb overestimation by slowing down learning. For the real data ratio, $\kappa = 0.05$, widely used in prior conservative algorithms (Yu et al., 2020), often yields poor performance. Large κ acts as a *softened* regularizer, limiting overtrust in the model while avoiding explicit penalties.

Ablation: Introducing explicit conservatism to NEUBAY helps some tasks, but not on average. We study explicit conservatism by penalizing imagined rewards with $\lambda \frac{U_{\boldsymbol{\theta}}(\hat{s},\hat{a})}{\mathbb{E}_{(s,a)\sim\mathcal{D}}[U_{\boldsymbol{\theta}}(s,a)]}$, normalized by the dataset average so that λ is more comparable across datasets. As summarized in Tab. 2, strong penalties ($\lambda = 1.0$ or 5.0) generally hurt performance, while a small penalty ($\lambda = 0.04$) performs comparably to NEUBAY ($\lambda = 0$). Effect remains dataset-dependent, as detailed in Tab. 13. Consistent with the bandit intuition, heavy penalties significantly worsen performance on 6 of 8 low-quality datasets, while leaving $\star$-medium-expert datasets largely unaffected. In contrast, some tasks benefit substantially: hopper-random-v2 (24.5 $\rightarrow$ 48.2, a new SOTA), hopper-medium-v2 (54.2 $\rightarrow$ 105.8), and pen-human-v1 (20.8 $\rightarrow$ 35.9), but each requires a

different λ, highlighting the need for tuning. However, penalties are not a universal remedy for narrow data: Walker2d-v3-Medium and halfcheetah-medium-v2 still fall short of the best baselines, and in AntMaze, penalties consistently harm performance.

5.3. Computation Costs

The total training cost of NEUBAY consists of two phases: *world-model training* and *agent training*. In practice, the world model is trained once per seed and its ensemble checkpoint is reused when tuning agents, making world-model training a *one-time cost*, while agent training dominates the overall computation. All experiments were run on a single NVIDIA L40S GPU (48 GB) with 3 seeds in parallel, using a fully vectorized JAX implementation.

For halfcheetah-medium-expert-v2 with 2M gradient steps and ensemble size $N = 100$, world-model training costs 6 hours per seed and 10.7 GB memory, while recurrent agent training costs 4.4 hours per seed and 2.6 GB memory. As detailed in Sec. F.6, the rollout inference cost is minimal, and increasing ensemble size N or rollout size K has only a minor effect on runtime.

6. Discussion

Practical guidelines for NEUBAY. Based on our experiments, we suggest the following guidelines when applying NEUBAY to new tasks or after modifying key modules:

- *When to use* NEUBAY*?* NEUBAY is best suited for low-quality or moderate-coverage datasets and settings that permit *test-time adaptation*. It may underperform on narrow-coverage ones possibly due to current limits in posterior modeling. In safety-critical domains, uncertainty penalties can be reintroduced to prioritize safety.
- *Defaults first.* Use a large ensemble size, LayerNorm in the world model, and uncertainty threshold $\zeta = 1.0$.
- *Encoder learning rate and real data ratio.* Tune η_ϕ within a wide range, typically 3–300$\times$ smaller than the MLP head's learning rate. Adjust κ within $[0.5, 0.95]$.
- *Monitor overestimation.* Mitigate it by reducing the discount factor γ or lowering the learning rate η_ϕ, as suggested by our analysis in Sec. 4.1.

Conclusion. We revisit (explicit) conservatism as the dominant principle in offline RL and show that it is not universally optimal. Instead, we advance a Bayesian perspective that trains history-dependent agents to maximize expected rewards over a posterior of world models. Building on this principle, we show that *long-horizon rollouts* play a critical role in reducing value overestimation once explicit conservatism is removed, and we propose key design choices that enable both performing and learning from such rollouts, yielding the NEUBAY algorithm. Across diverse benchmarks, NEUBAY is competitive with conservative baselines and particularly effective on low-quality and moderate-coverage datasets. More broadly, our work suggests a shift in offline RL toward the *era of experience* (Silver & Sutton, 2025), where agents are designed to adapt and refine behavior as uncertainty resolves through test-time interaction.

Future work. We see several promising directions for future work. Improving world models, through multi-step prediction or generative models, can help push the limits of Bayesian offline RL. Better uncertainty quantification remains key for planning, suggesting deeper connections to Bayesian inference are worth exploring. Finally, reducing sensitivity to hyperparameters and dataset characteristics, as well as replacing current test-environment-based tuning with offline policy selection, would make NEUBAY more robust and practical; since NEUBAY already fits a Bayesian world model, a natural direction is to carry out such validation under the learned posterior (Fellows et al., 2025).

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 which we feel must be specifically highlighted here.

References

Agarwal, A., Kakade, S., Krishnamurthy, A., and Sun, W. Flambe: Structural complexity and representation learning of low rank mdps. *Advances in neural information processing systems*, 33: 20095–20107, 2020a. 28

Agarwal, R., Schuurmans, D., and Norouzi, M. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning*, pp. 104–114. PMLR, 2020b. 17

An, G., Moon, S., Kim, J.-H., and Song, H. O. Uncertainty-based offline reinforcement learning with diversified q-ensemble. *Advances in neural information processing systems*, 34:7436–7447, 2021. 6, 16, 17, 20

Argenson, A. and Dulac-Arnold, G. Model-based offline planning. In *International Conference on Learning Representations*, 2020. 16

Asadi, K., Misra, D., and Littman, M. Lipschitz continuity in model-based reinforcement learning. In *International conference on machine learning*, pp. 264–273. PMLR, 2018. 18

Asadi, K., Misra, D., Kim, S., and Littman, M. L. Combating the compounding-error problem with a multi-step model. *arXiv preprint arXiv:1905.13320*, 2019. 18

Ba, J. L., Kiros, J. R., and Hinton, G. E. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 2, 5, 18

Bai, C., Wang, L., Yang, Z., Deng, Z.-H., Garg, A., Liu, P., and Wang, Z. Pessimistic bootstrapping for uncertainty-driven of-

fline reinforcement learning. In *International Conference on Learning Representations*, 2022. 20

Ball, P. J., Smith, L., Kostrikov, I., and Levine, S. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, pp. 1577–1594. PMLR, 2023. 5

Beck, J., Vuorio, R., Liu, E. Z., Xiong, Z., Zintgraf, L., Finn, C., and Whiteson, S. A survey of meta-reinforcement learning. *arXiv preprint arXiv:2301.08028*, 2023. 18

Bhardwaj, M., Xie, T., Boots, B., Jiang, N., and Cheng, C.-A. Adversarial model for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36:1245–1269, 2023. 6, 16

Bickel, P., Diggle, P., Fienberg, S., Gather, U., Olkin, I., and Zeger, S. Springer series in statistics. *Principles and Theory for Data Mining and Machine Learning. Cham, Switzerland: Springer*, 6, 2009. 27

Blelloch, G. E. Prefix sums and their applications. *School of Computer Science, Carnegie Mellon University Pittsburgh, PA, USA*, 1990. 19

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/jax-ml/jax. 31, 34

Cassandra, A. R., Kaelbling, L. P., and Littman, M. L. Acting optimally in partially observable stochastic domains. In *Aaai*, volume 94, pp. 1023–1028, 1994. 21

Chen, J., Xu, L., Chen, W., and Schneider, J. Bayes adaptive monte carlo tree search for offline model-based reinforcement learning. In *International Conference on Learning Representations*, 2026. 17

Chen, L., Lu, K., Rajeswaran, A., Lee, K., Grover, A., Laskin, M., Abbeel, P., Srinivas, A., and Mordatch, I. Decision transformer: Reinforcement learning via sequence modeling. *Advances in neural information processing systems*, 34:15084–15097, 2021a. 2

Chen, X., Wang, C., Zhou, Z., and Ross, K. W. Randomized ensembled double q-learning: Learning fast without a model. In *International Conference on Learning Representations*, 2021b. 34, 49

Chen, X., Wang, S., McAuley, J., Jannach, D., and Yao, L. On the opportunities and challenges of offline reinforcement learning for recommender systems. *ACM Transactions on Information Systems*, 42(6):1–26, 2024. 1

Chen, X., Wang, G., Yan, K., and Zhao, L. Vipo: Value function inconsistency penalized offline reinforcement learning. *arXiv preprint arXiv:2504.11944*, 2025. 6, 16

Chen, X.-H., Yu, Y., Li, Q., Luo, F.-M., Qin, Z., Shang, W., and Ye, J. Offline model-based adaptable policy learning. *Advances in Neural Information Processing Systems*, 34:8432–8443, 2021c. 2, 6, 16, 38

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 19

Choi, Y., Zhao, L., Zhang, C., Song, L., Bian, J., and Kim, K.-E. Diversification of adaptive policy for effective offline reinforcement learning. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pp. 3863–3871, 2024. 6, 16, 38

Chua, K., Calandra, R., McAllister, R., and Levine, S. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018. 3

Dearden, R., Friedman, N., Russell, S., et al. Bayesian q-learning. *Aaai/iaai*, 1998:761–768, 1998. 18

Derman, E., Mankowitz, D. J., Mann, T. A., and Mannor, S. Soft-robust actor-critic policy-gradient. In *Uncertainty in Artificial Intelligence*, 2018. 19

Derman, E., Mankowitz, D., Mann, T., and Mannor, S. A bayesian approach to robust reinforcement learning. In *Uncertainty in Artificial Intelligence*, pp. 648–658. PMLR, 2020. 18

Donsker, M. D. and Varadhan, S. S. Asymptotic evaluation of certain markov process expectations for large time, i. *Communications on pure and applied mathematics*, 28(1):1–47, 1975. 21

Dorfman, R., Shenfeld, I., and Tamar, A. Offline meta reinforcement learning–identifiability challenges and effective data collection strategies. *Advances in Neural Information Processing Systems*, 34:4607–4618, 2021. 18

Duff, M. O. *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes*. University of Massachusetts Amherst, 2002. 1, 3, 16, 18

Dulac-Arnold, G., Levine, N., Mankowitz, D. J., Li, J., Paduraru, C., Gowal, S., and Hester, T. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning*, 110(9):2419–2468, 2021. 1

Ellsberg, D. Risk, ambiguity, and the savage axioms. *The quarterly journal of economics*, 75(4):643–669, 1961. 3, 18

Ernst, D., Geurts, P., and Wehenkel, L. Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, 6, 2005. 17

Feinberg, V., Wan, A., Stoica, I., Jordan, M. I., Gonzalez, J. E., and Levine, S. Model-based value estimation for efficient model-free reinforcement learning. In *International Conference on Machine Learning*, 2018. 17

Fellows, M., Wibault, C., Berdica, U., Forkel, J., Foerster, J. N., and Osborne, M. A. Sorel and torel: Two methods for fully offline reinforcement learning. *arXiv preprint arXiv:2505.22442*, 2025. 3, 9, 30

Fort, S., Hu, H., and Lakshminarayanan, B. Deep ensembles: A loss landscape perspective. *arXiv preprint arXiv:1912.02757*, 2019. 3

Frauenknecht, B., Eisele, A., Subhasish, D., Solowjow, F., and Trimpe, S. Trust the model where it trusts itself-model-based actor-critic with uncertainty-aware rollout adaption. In *International Conference on Machine Learning*, pp. 13973–14005. PMLR, 2024. 2, 5, 18

Frauenknecht, B., Subhasish, D., Solowjow, F., and Trimpe, S. On rollouts in model-based reinforcement learning. In *International Conference on Learning Representations*, 2025. 18

Fu, J., Kumar, A., Nachum, O., Tucker, G., and Levine, S. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020. 1, 2, 6, 31

Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pp. 2052–2062. PMLR, 2019. 2, 3, 4, 17, 19, 20

Gal, Y. *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge, 2016. 3

Ghasemipour, S. K. S., Schuurmans, D., and Gu, S. S. Emaq: Expected-max q-learning operator for simple yet effective offline and online rl. In *International Conference on Machine Learning*, pp. 3682–3691. PMLR, 2021. 20

Ghavamzadeh, M., Mannor, S., Pineau, J., Tamar, A., et al. Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 8(5-6):359–483, 2015. 3, 18

Ghosh, D., Rahme, J., Kumar, A., Zhang, A., Adams, R. P., and Levine, S. Why generalization in rl is difficult: Epistemic pomdps and implicit partial observability. *Advances in neural information processing systems*, 34:25502–25515, 2021. 18

Ghosh, D., Ajay, A., Agrawal, P., and Levine, S. Offline rl policies should be trained to be adaptive. In *International Conference on Machine Learning*, pp. 7513–7530. PMLR, 2022. 1, 2, 3, 6, 16, 18, 21

Gu, A. and Dao, T. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 19

Gulcehre, C., Wang, Z., Novikov, A., Paine, T., Gómez, S., Zolna, K., Agarwal, R., Merel, J. S., Mankowitz, D. J., Paduraru, C., et al. Rl unplugged: A suite of benchmarks for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:7248–7259, 2020. 1, 17

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. Pmlr, 2018a. 31, 34, 49

Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., Kumar, V., Zhu, H., Gupta, A., Abbeel, P., et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018b. 34

Hafner, D., Pasukonis, J., Ba, J., and Lillicrap, T. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023. 18

Hansen, N., Su, H., and Wang, X. Td-mpc2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024. 18

Hochreiter, S. and Schmidhuber, J. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 18

Hong, J., Kumar, A., and Levine, S. Confidence-conditioned value functions for offline reinforcement learning. In *International Conference on Learning Representations*, 2023. 17

Janner, M., Fu, J., Zhang, M., and Levine, S. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32, 2019. 2, 4, 5, 7, 15, 17, 18

Janner, M., Li, Q., and Levine, S. Offline reinforcement learning as one big sequence modeling problem. *Advances in neural information processing systems*, 34:1273–1286, 2021. 16

Jeong, J., Wang, X., Gimelfarb, M., Kim, H., Abdulhai, B., and Sanner, S. Conservative bayesian model-based value expansion for offline policy optimization. In *International Conference on Learning Representations*, 2023. 2, 6, 17, 20

Jiang, Y., Kolter, J. Z., and Raileanu, R. On the importance of exploration for generalization in reinforcement learning. *Advances in Neural Information Processing Systems*, 36:12951–12986, 2023. 18

Jin, Y., Yang, Z., and Wang, Z. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, pp. 5084–5096. PMLR, 2021. 1, 30

Kapturowski, S., Ostrovski, G., Quan, J., Munos, R., and Dabney, W. Recurrent experience replay in distributed reinforcement learning. In *International conference on learning representations*, 2018. 16

Kidambi, R., Rajeswaran, A., Netrapalli, P., and Joachims, T. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33:21810–21823, 2020. 16, 20

Kidger, P. and Garcia, C. Equinox: neural networks in JAX via callable PyTrees and filtered transformations. *Differentiable Programming workshop at Neural Information Processing Systems 2021*, 2021. 31

Kim, B. and Oh, M.-h. Model-based offline reinforcement learning with count-based conservatism. In *International Conference on Machine Learning*, pp. 16728–16746. PMLR, 2023. 16

Kostrikov, I., Nair, A., and Levine, S. Offline reinforcement learning with implicit q-learning. In *International Conference on Learning Representations*, 2022. 6, 19, 20, 34

Kumar, A., Fu, J., Soh, M., Tucker, G., and Levine, S. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019. 4, 19, 20

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020. 5, 6

Lagoudakis, M. G. and Parr, R. Least-squares policy iteration. *The Journal of Machine Learning Research*, 4:1107–1149, 2003. 17

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017. 3

Lambert, N., Pister, K., and Calandra, R. Investigating compounding prediction errors in learned dynamics models. *arXiv preprint arXiv:2203.09637*, 2022. 2, 17

Lange, S., Gabel, T., and Riedmiller, M. Batch reinforcement learning. In *Reinforcement learning: State-of-the-art*, pp. 45–73. Springer, 2012. 1

Levin, D. A. and Peres, Y. *Markov chains and mixing times*, volume 107. American Mathematical Soc., 2017. 29

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. 1

Li, G., Shi, L., Chen, Y., Chi, Y., and Wei, Y. Settling the sample complexity of model-based offline reinforcement learning. *The Annals of Statistics*, 52(1):233–260, 2024. 22, 26, 30

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 *International Conference on Learning Representations*, 2025. 4, 6, 18, 19, 33, 34, 38

Lin, H., Xiao, S., Li, Y.-C., Zhang, Z., Sun, Y., Jia, C., and Yu, Y. ADM-v2: Pursuing full-horizon roll-out in dynamics models for offline policy learning and evaluation. In *International Conference on Learning Representations*, 2026. 19

Liu, Z., Fang, Z., Lyu, J., and Li, X. Leveraging score-based models for generating penalization in model-based offline reinforcement learning. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, pp. 1389–1398, 2025. 6

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 33

Lu, C., Ball, P., Parker-Holder, J., Osborne, M., and Roberts, S. J. Revisiting design choices in offline model based reinforcement learning. In *International Conference on Learning Representations*, 2021. 1, 18, 43

Lu, C., Schroecker, Y., Gu, A., Parisotto, E., Foerster, J., Singh, S., and Behbahani, F. Structured state space models for in-context reinforcement learning. *Advances in Neural Information Processing Systems*, 36:47016–47031, 2023. 19, 34

Lu, C., Shi, R., Liu, Y., Hu, K., Du, S. S., and Xu, H. Rethinking transformers in solving pomdps. In *Forty-first International Conference on Machine Learning*, 2024. 19

Luis, C. E., Bottero, A. G., Vinogradska, J., Berkenkamp, F., and Peters, J. Uncertainty representations in state-space layers for deep reinforcement learning under partial observability. *arXiv preprint arXiv:2409.16824*, 2024. 19

Luo, F.-M., Tu, Z., Huang, Z., and Yu, Y. Efficient recurrent off-policy rl requires a context-encoder-specific learning rate. *Advances in Neural Information Processing Systems*, 37:48484–48518, 2024a. 2, 6, 8, 19, 34, 35

Luo, F.-M., Xu, T., Cao, X., and Yu, Y. Reward-consistent dynamics models are strongly generalizable for offline reinforcement learning. In *International Conference on Learning Representations*, 2024b. 4, 16, 18

Ma, L., Liang, H., Qiang, M., Tang, L., Ma, X., Wong, Z. H., Niu, J., Shen, C., He, R., Li, Y., Zhang, W., and CUI, B. Learning what reinforcement learning can't: Interleaved online fine-tuning for hardest questions. In *International Conference on Learning Representations*, 2026. 1

Ma, M., Ni, T., Gehring, C., D'Oro, P., and Bacon, P.-L. Do transformer world models give better policy gradients? In *International Conference on Machine Learning*, pp. 33855–33879. PMLR, 2024. 18

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013. 31

Moeini, A., Wang, J., Beck, J., Blaser, E., Whiteson, S., Chandra, R., and Zhang, S. A survey of in-context reinforcement learning. *arXiv preprint arXiv:2502.07978*, 2025. 17

Morad, S., Kortvelesy, R., Bettini, M., Liwicki, S., and Prorok, A. Popgym: Benchmarking partially observable reinforcement learning. In *The Eleventh International Conference on Learning Representations*, 2023. 19

Morad, S., Lu, C., Kortvelesy, R., Liwicki, S., Foerster, J., and Prorok, A. Recurrent reinforcement learning with memoroids. *Advances in Neural Information Processing Systems*, 37:14386–14416, 2024. 2, 5, 19, 34

Munos, R. and Szepesvári, C. Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, 9(5), 2008. 30

Nair, A., Gupta, A., Dalal, M., and Levine, S. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020. 17

Ni, T., Eysenbach, B., and Salakhutdinov, R. Recurrent model-free rl can be a strong baseline for many pomdps. In *International Conference on Machine Learning*, pp. 16691–16723. PMLR, 2022. 5, 18, 19, 34

Ni, T., Ma, M., Eysenbach, B., and Bacon, P.-L. When do transformers shine in rl? decoupling memory from credit assignment. In *Advances in Neural Information Processing Systems*, 2023. 2, 5, 19

Orvieto, A., Smith, S. L., Gu, A., Fernando, A., Gulcehre, C., Pascanu, R., and De, S. Resurrecting recurrent neural networks for long sequences. In *International Conference on Machine Learning*, pp. 26670–26698. PMLR, 2023. 5, 19, 34, 35

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. Deep exploration via bootstrapped dqn. *Advances in neural information processing systems*, 29, 2016. 18

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019. 3

Pan, F., He, J., Tu, D., and He, Q. Trust the model when it is confident: Masked model-based actor-critic. *Advances in neural information processing systems*, 33:10537–10546, 2020. 5, 18

Parisotto, E., Song, F., Rae, J., Pascanu, R., Gulcehre, C., Jayakumar, S., Jaderberg, M., Kaufman, R. L., Clark, A., Noury, S., et al. Stabilizing transformers for reinforcement learning. In *International conference on machine learning*, pp. 7487–7498. PMLR, 2020. 19

Park, K. and Lee, Y. Model-based offline reinforcement learning with lower expectile q-learning. In *International Conference on Learning Representations*, 2025. 6, 16, 33, 38

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. 1

Puterman, M. L. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014. 2

Qiao, Z., Lyu, J., Jiao, K., Liu, Q., and Li, X. Sumo: Search-based uncertainty estimation for model-based offline reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20033–20041, 2025. 6, 16

Qiao, Z., Lyu, J., Lyu, B., Shu, Y., Gao, S., and Qiu, S. Model-based offline RL via robust value-aware model learning with implicitly differentiable adaptive weighting. In *International Conference on Learning Representations*, 2026. 6

Qin, R.-J., Zhang, X., Gao, S., Chen, X.-H., Li, Z., Zhang, W., and Yu, Y. Neorl: A near real-world benchmark for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 35:24753–24765, 2022. 2, 6, 32, 38

Rajeswaran, A., Ghotra, S., Ravindran, B., and Levine, S. Epopt: Learning robust neural network policies using model ensembles. In *International Conference on Learning Representations*, 2017. 18

Rashidinejad, P., Zhu, B., Ma, C., Jiao, J., and Russell, S. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information Processing Systems*, 34:11702–11716, 2021. 30

Riedmiller, M. Neural fitted q iteration–first experiences with a data efficient neural reinforcement learning method. In *Machine Learning: ECML 2005: 16th European Conference on Machine Learning, Porto, Portugal, October 3-7, 2005. Proceedings 16*, pp. 317–328. Springer, 2005. 17

Rigter, M., Lacerda, B., and Hawes, N. Risk-averse bayes-adaptive reinforcement learning. *Advances in Neural Information Processing Systems*, 34:1142–1154, 2021. 18

Rigter, M., Lacerda, B., and Hawes, N. Rambo-rl: Robust adversarial model-based offline reinforcement learning. *Advances in neural information processing systems*, 35:16082–16097, 2022. 3, 6, 16, 20

Rigter, M., Lacerda, B., and Hawes, N. One risk to rule them all: A risk-sensitive perspective on model-based offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2023. 16, 21

Schrittwieser, J., Hubert, T., Mandhane, A., Barekatain, M., Antonoglou, I., and Silver, D. Online and offline reinforcement learning by planning with a learned model. *Advances in Neural Information Processing Systems*, 34:27580–27591, 2021. 17

Sentenac, F., Lee, I., and Szepesvari, C. Balancing optimism and pessimism in offline-to-online learning. *arXiv preprint arXiv:2502.08259*, 2025. 17

Silver, D. and Sutton, R. S. Welcome to the era of experience. *preprint*, 2025. 9

Sims, A., Lu, C., Foerster, J. N., and Teh, Y. W. The edge-of-reach problem in offline model-based reinforcement learning. *Advances in Neural Information Processing Systems*, 37:63029–63056, 2024. 4, 17, 21, 22

Srinivasan, P. and Knottenbelt, W. Offline model-based reinforcement learning with anti-exploration. *arXiv preprint arXiv:2408.10713*, 2024. 6, 16

Sun, Y. Offlinerl-kit: An elegant pytorch offline reinforcement learning library. https://github.com/yihaosun1124/OfflineRL-Kit, 2023. 33

Sun, Y., Zhang, J., Jia, C., Lin, H., Ye, J., and Yu, Y. Model-bellman inconsistency for model-based offline reinforcement learning. In *International Conference on Machine Learning*, pp. 33177–33194. PMLR, 2023. 6, 16, 20, 21, 33, 34, 38

Sutton, R. S. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Machine learning proceedings 1990*, pp. 216–224. Elsevier, 1990. 15

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018. 1, 15

Swazinna, P., Udluft, S., and Runkler, T. User-interactive offline reinforcement learning. In *International Conference on Learning Representations*, 2023. 17

Talvitie, E. Model regularization for stable sample rollouts. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, pp. 780–789, 2014. 17

Tarasov, D., Kurenkov, V., Nikulin, A., and Kolesnikov, S. Revisiting the minimalist approach to offline reinforcement learning. *Advances in Neural Information Processing Systems*, 36:11592–11620, 2023a. 6

Tarasov, D., Nikulin, A., Akimov, D., Kurenkov, V., and Kolesnikov, S. Corl: Research-oriented deep offline reinforcement learning library. *Advances in Neural Information Processing Systems*, 36:30997–31020, 2023b. 38

Uehara, M. and Sun, W. Pessimistic model-based offline reinforcement learning under partial coverage. In *International Conference on Learning Representations*, 2022. 1, 3, 18, 20, 22, 23, 28, 30

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 19

Vlassis, N., Ghavamzadeh, M., Mannor, S., and Poupart, P. Bayesian reinforcement learning. *Reinforcement Learning: State-of-the-Art*, pp. 359–386, 2012. 18

Vuorio, R., Fellows, M., Lu, C., Grislain, C., and Whiteson, S. A bayesian solution to the imitation gap. *arXiv preprint arXiv:2407.00495*, 2024. 18

Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., and Freitas, N. Dueling network architectures for deep reinforcement learning. In *International conference on machine learning*, pp. 1995–2003. PMLR, 2016. 34

Wiesemann, W., Kuhn, D., and Rustem, B. Robust markov decision processes. *Mathematics of Operations Research*, 38(1): 153–183, 2013. 3, 19, 20

Wilson, A. G. and Izmailov, P. Bayesian deep learning and a probabilistic perspective of generalization. *Advances in neural information processing systems*, 33:4697–4708, 2020. 3

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 *International Conference on Learning Representations*, 2023. 19

Yang, S., Zhang, S., Feng, Y., and Zhou, M. A unified framework for alternating offline model training and policy learning. *Advances in Neural Information Processing Systems*, 35:17216–17232, 2022. 16

Yang, Z. and Nguyen, H. Recurrent off-policy baselines for memory-based continuous control. *arXiv preprint arXiv:2110.12628*, 2021. 18

Yarats, D., Brandfonbrener, D., Liu, H., Laskin, M., Abbeel, P., Lazaric, A., and Pinto, L. Don't change the algorithm, change the data: Exploratory data for offline reinforcement learning. In *ICLR 2022 Workshop on Generalizable Policy Learning in Physical World*, 2022. 17

Young, K. J., Ramesh, A., Kirsch, L., and Schmidhuber, J. The benefits of model-based generalization in reinforcement learning. In *International Conference on Machine Learning*, pp. 40254–40276. PMLR, 2023. 4

Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J. Y., Levine, S., Finn, C., and Ma, T. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020. 2, 3, 6, 8, 16, 17, 18, 20, 21, 31, 33

Yu, T., Kumar, A., Rafailov, R., Rajeswaran, A., Levine, S., and Finn, C. Combo: Conservative offline model-based policy optimization. *Advances in neural information processing systems*, 34:28954–28967, 2021. 6, 16

Zhai, Y., Li, Y., Gao, Z., Gong, X., Xu, K., Feng, D., Bo, D., and Wang, H. Optimistic model rollouts for pessimistic offline policy optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16678–16686, 2024. 17

Zhan, X., Zhu, X., and Xu, H. Model-based offline planning with trajectory pruning. *arXiv preprint arXiv:2105.07351*, 2021. 2, 5, 16, 18

Zhang, J., Lyu, J., Ma, X., Yan, J., Yang, J., Wan, L., and Li, X. Uncertainty-driven trajectory truncation for data augmentation in offline reinforcement learning. In *ECAI 2023*, pp. 3018–3025. IOS Press, 2023. 5, 6, 18

Zhang, J., Fang, L., Shi, K., Wang, W., and Jing, B. Q-distribution guided q-learning for offline reinforcement learning: Uncertainty penalized q-value via consistency model. *Advances in Neural Information Processing Systems*, 37:54421–54462, 2024a. 32

Zhang, R., Hu, Y., and Li, N. Soft robust mdps and risk-sensitive mdps: Equivalence, policy gradient, and sample complexity. In *The Twelfth International Conference on Learning Representations*, 2024b. 1, 3, 19, 21

Zhou, G., Swaminathan, S., Raju, R. V., Guntupalli, J. S., Lehrach, W., Ortiz, J., Dedieu, A., Lázaro-Gredilla, M., and Murphy, K. Diffusion model predictive control. *Transactions on Machine Learning Research*, 2025. 43

Zintgraf, L., Shiarlis, K., Igl, M., Schulze, S., Gal, Y., Hofmann, K., and Whiteson, S. Varibad: A very good method for bayes-adaptive deep rl via meta-learning. In *International Conference on Learning Representations*, 2020. 3, 18, 19

Contents

A. Related Work

A.1. Offline Conservative Model-Based Methods

Model-based methods can be broadly grouped into background planning and decision-time planning (Sutton & Barto, 2018, Chapter 8.8). Background planning, such as Dyna-style methods (Sutton, 1990; Janner et al., 2019), uses model-generated rollouts to learn a global policy or value function, which is then queried for action selection. Decision-time planning, such as model predictive control (MPC), performs online look-ahead at test time to select actions for the current state. Our work belongs to *background planning*. Below, we review these paradigms within the offline setting, focusing on how they incorporate explicit conservatism.

Conservative background planning algorithms. In this category of work, most methods enforce conservatism by constructing a *pessimistic MDP* that penalizes imagined state-action pairs, which we refer to as **uncertainty-penalized**

pessimism in Sec. B.1. The earliest examples are MOReL (Kidambi et al., 2020) and MOPO (Yu et al., 2020), both approximating the uncertainty set with world ensembles. MOReL adopts strong pessimism by mapping any state-action pair with ensemble disagreement above a threshold to an absorbing state with a large penalty. This aggressive design disables value bootstrapping on uncertain regions and explains why MOReL supports long-horizon planning (up to 500 steps) even with severe compounding errors. In contrast, MOPO applies a milder penalty to imagined rewards based on the aleatoric uncertainty, retaining bootstrapping and thus limiting rollouts to very short horizons (typically 1-5 steps).

Building on MOPO, several follow-up works redesign the uncertainty quantifier: MOBILE (Sun et al., 2023) uses inconsistencies in Bellman operators across ensemble members; Kim & Oh (2023) uses the inverse frequencies of state-action pairs; MoMo (Srinivasan & Knottenbelt, 2024) adopts energy-based models; SUMO (Qiao et al., 2025) employs k-nearest neighbors. Other works introduce alternative conservative mechanisms: COMBO (Yu et al., 2021) uses CQL regularizer; LEQ (Park & Lee, 2025) uses lower expectile regression. From the model side, several works improve the model learning or rollout sampling: Luo et al. (2024b) trains discriminators on (s, a, s') to resample model rollouts by fidelity; VIPO (Chen et al., 2025) augments the standard MLE loss with a value-consistency objective that aligns behavior-policy values under model and true dynamics.

A different line of work jointly trains adversarial world models and policies, rather than freezing the models after pretraining, which we refer to as **adversarial model-based pessimism** in Sec. B.1. Several adversarial objectives have been explored: RAMBO (Rigter et al., 2022) minimizes value estimates; Yang et al. (2022) minimizes the divergence between real and imagined state-action distributions; ARMOR (Bhardwaj et al., 2023) minimizes value differences between the current and a reference policy; Rigter et al. (2023) extends the uncertainty set to incorporate aleatoric uncertainty.

Our work differs from these prior lines of research in principle, core components, and algorithmic design. We replace conservatism with Bayesianism as the guiding principle, substituting the standard Markov actor-critic with a history-dependent one required by Bayesian principle, and derive the NEUBAY algorithm driven by long-horizon planning, fundamentally distinct from prior methods. Moreover, prior works typically rely on tuning two critical hyperparameters: the *conservatism coefficient* λ and *rollout horizon* H, to achieve strong performance. In contrast, NEUBAY eliminates the need for λ by design and replaces H with an uncertainty quantile ζ, which remains fixed at 1.0 in main experiments.

Conservative decision-time planning algorithms. In this line of work, a planner is used at test time to select actions from the current state. The planner is composed of a world model, a conservative policy obtained by behavior cloning (Argenson & Dulac-Arnold, 2020), and a conservative value function learned by fitted Q evaluation (Zhan et al., 2021) or other pessimistic estimators (Janner et al., 2021). Given the conservative policy as a prior, the planner samples exploratory trajectories from world models and chooses the action with the highest estimated value. Decision-time planning can discover better actions at test time by performing targeted exploration from a specific state, particularly useful in unseen states. It is distinct from classical RL because it does not learn a global policy to maximize returns; instead, it directly searches for actions.

A.2. Offline Bayesian-Inspired and Non-Conservative RL

Bayesian-inspired algorithms. Several offline RL works draw inspiration from Bayesian ideas, such as using model posteriors and connections to Bayes-adaptive MDPs (Duff, 2002). However, their formulations or algorithms typically differ from optimizing the epistemic POMDP in Eq. 2, which is the focus of our work. For example, although Ghosh et al. (2022) introduces the Bayesian model-based formulation in Eq. 2, their proposed APE-V algorithm adopts a *conservative model-free* approximation. Rather than maintaining a posterior over dynamics models, APE-V uses an ensemble of belief-state Q-functions as a surrogate posterior over values, trained purely via TD errors. Each Q-function is trained using SAC-N (An et al., 2021), which enforces conservatism by minimizing over Q-ensemble predictions.

Similar to our algorithm, **MAPLE** (Chen et al., 2021c) and **MoDAP** (Choi et al., 2024) learn an ensemble of models for recurrent policy training. Unlike our approach, both algorithms perform short-horizon rollouts (≤ 20 steps). Moreover, both store the hidden states of recurrent policies in the replay buffer and reuse them to initialize the policy's context for rollouts and updates. This design is known to induce context staleness in recurrent RL (Kapturowski et al., 2018), whereas our algorithm avoids this issue by storing and sampling full histories directly. MAPLE also reintroduces an uncertainty penalty and terminates rollouts based on a predefined state boundary, making it a conservative method. MoDAP remains penalty-free but fine-tunes the world models during policy learning to maintain ensemble diversity; therefore, its objective departs from the epistemic POMDP, which requires freezing the posterior during policy optimization. In contrast, our work aims to stay as close as possible to the epistemic POMDP: we avoid explicit conservatism, develop design choices for long-horizon rollouts, and do not fine-tune models during policy learning.

Other works are Bayesian-inspired in different ways. CBOP (Jeong et al., 2023) uses a model posterior to weight multi-step TD targets in an MVE-style update (Feinberg et al., 2018) and applies lower-confidence penalties. BA-MCTS (Chen et al., 2026) proposes a Bayesian Monte Carlo planning method with an uncertainty penalty, used as a policy-improvement operator.

Offline RL algorithms without explicit conservatism. Although conservatism dominates modern offline RL, classic offline (batch) RL was originally developed without any conservatism (Lagoudakis & Parr, 2003; Ernst et al., 2005; Riedmiller, 2005). These methods are based on *fitted Q-iteration*, which directly applies Markov Q-learning to offline data. While effective on small-scale problems with sufficient data coverage (Riedmiller, 2005), such algorithms are known to fail in high-dimensional settings (Fujimoto et al., 2019). More recently, Agarwal et al. (2020b) provides an *optimistic* perspective, showing that standard off-policy RL trained on the 50M transitions from DQN's replay buffer can outperform the behavior policy. Likewise, Yarats et al. (2022) demonstrates that off-policy RL can surpass conservative methods on diverse-coverage datasets collected by unsupervised agents.

In the model-based setting, MBPO (Janner et al., 2019), using short-horizon rollouts without conservatism, is known to underperform conservative counterparts such as MOPO in offline RL benchmarks (Yu et al., 2020). However, MuZero Unplugged (Schrittwieser et al., 2021) shows that MuZero, without conservatism, can achieve strong performance on the RL Unplugged suite (Gulcehre et al., 2020), which contains 200M Atari transitions and DMC control datasets using replay buffer from near-optimal RL agents. Zhai et al. (2024) observes that making MOPO optimistic instead of pessimistic can achieve strong performance on halfcheetah-random-v2, but worse on other D4RL tasks.

In summary, prior work on offline RL without explicit conservatism largely relies on standard off-policy RL, which treats offline RL as optimizing a single MDP and thus follows an *optimistic* principle (Agarwal et al., 2020b). Such optimism can work well when the dataset has broad state-action coverage but typically fails under limited coverage. Our approach takes a different non-conservative path: instead of being optimistic, it follows a neutral Bayesian principle that lies **between optimism and pessimism**. This allows NEUBAY to avoid the failure modes of optimistic methods and, for the first time to our knowledge, extends the effectiveness of non-conservative offline RL to *low-quality* and *moderate-coverage* datasets, while still benefiting from diverse coverage when available.

Test-time adaptation and offline-to-online RL. Finally, our method performs test-time adaptation, but it differs from prior approaches in its mechanism. For example, Hong et al. (2023); Swazinna et al. (2023) adapt at deployment by dynamically tuning the degree of conservatism without retraining. In contrast, our Bayes-adaptivity does not adjust hyperparameters or conservatism at test time. Instead, it arises purely from the policy's in-context memory: by conditioning on observed history, the recurrent policy maintains an implicit belief about the true MDP, adapting its behavior intra-episode without any parameter updates.

While one could view in-context memory as inducing an implicit policy improvement (Moeini et al., 2025), we do not treat this mechanism as explicit online learning in this work (i.e., offline-to-online RL (Nair et al., 2020)). In offline-to-online setting, Sentenac et al. (2025) analyzes bandit problems and shows that optimism can be preferable to pessimism during online learning, depending on the available interaction budget. While their focus is on a different regime, this observation is conceptually related to our finding that pessimism can hinder online adaptation.

A.3. Model-Generated Rollouts in RL

Effect of rollout length on value estimation. Sims et al. (2024) identifies the *edge-of-reach* problem in offline conservative model-based RL (MBRL): when short-horizon rollouts are used, bootstrapping often occurs on states whose Q-values are never directly trained, leading to substantial overestimation once uncertainty penalties vanish (e.g., under true dynamics). Closely related to our work, they emphasize that such overestimation induced by short rollouts constitutes a distinct failure mode in offline MBRL, separate from classic compounding errors. To mitigate this, they replace dynamics-based uncertainty penalties with value-based ones via pessimistic Q-ensembles (An et al., 2021), while still relying on short-horizon rollouts.

Our work extends this line of reasoning in two ways. (1) We show that the same overestimation mechanism arises under Bayesian Bellman backups and extend Sims et al. (2024, Proposition 1) to Bayesian setting in Sec. C. (2) Instead of conservative short-horizon rollouts, we use adaptive long-horizon rollouts that exploit the vanishing factor γ^H to naturally reduce the bootstrapped error and remove the need for conservatism.

Reducing compounding errors and the scale of rollout horizon. Model-generated rollouts suffer from compounding prediction errors (Talvitie, 2014; Lambert et al., 2022), which can grow quickly with the rollout horizon. These errors cause

the performance gap between the policy evaluated under the learned model and under the true MDP, as formalized by theory such as the simulation lemma (see Uehara & Sun (2022, Lemma 9) and our Lemma 4). To prevent large compounding errors from harming policy learning, most online and offline MBRL methods (Janner et al., 2019; Yu et al., 2020; Lu et al., 2021; Hafner et al., 2023; Hansen et al., 2024) restrict rollouts to very short horizons (typically **1-20 steps**). These approaches often share a minimalist setup: standard MLP world models without layer normalization, pure MLE training, and rollout procedures without uncertainty awareness.

Prior works attempt to reduce compounding errors and thus enable longer rollouts along three ways: *model architecture*, *training objective*, and *inference strategy*. On the architectural side, improving model smoothness can enhance generalization on unseen states (Asadi et al., 2018). On the training-objective side, multi-step architecture predicts future states given an initial state and an action sequence (Asadi et al., 2019), enabling horizons up to 500 steps with Transformers in online RL (Ma et al., 2024) and up to 50 steps with RNNs in offline RL (Lin et al., 2025). For inference strategies, MOREC (Luo et al., 2024b) resamples model rollouts using a fidelity estimator and scales horizons to 100 steps.

NEUBAY keeps the training objective fixed to standard one-step MLE, while contributing simple and effective components along the other two dimensions. Architecturally, we apply layer normalization (Ba et al., 2016) to stabilize prediction magnitudes (Sec. 4.3), increasing model smoothness (Asadi et al., 2018) and thus generalization; at inference time, we truncate rollouts using an uncertainty threshold as a proxy for compounding error (Sec. 4.2). Together, these components enable NEUBAY to successfully use horizons of **64-512** steps, which is rare in MBRL literature.

Mechanism of adaptive horizon. As discussed in Sec. 4.2, several prior works have used uncertainty thresholds to adaptively truncate rollouts. In the online RL setting, M2AC (Pan et al., 2020) truncates rollouts for states whose uncertainty ranks in the top 25% of the current-step batch and caps the horizon at 10. Similarly, MACURA (Frauenknecht et al., 2024) computes a 95% uncertainty quantile from the batch of first-step predictions in current rollouts, also with a maximum horizon of 10. Infoprop (Frauenknecht et al., 2025) extends MACURA by using accumulated uncertainty along the rollout as an additional truncation criterion and increases the maximum horizon to around 50. In the offline RL setting, MOPP (Zhan et al., 2021) adopts an 85% uncertainty quantile from the offline dataset to filter rollouts and imposes a maximum horizon of 16. TATU (Zhang et al., 2023) uses truncation thresholds proportional to the maximum uncertainty in the offline dataset, with a coefficient of 0.5 and a maximum horizon of 5.

A key difference is the *role* of truncation. Prior work uses adaptive horizons conservatively to keep rollouts short and limit compounding model error. In contrast, NEUBAY is motivated by the observation that, once explicit conservatism is removed, *long horizons become necessary* to reduce overestimation by reducing reliance on bootstrapping. Thus, we use uncertainty-based truncation to preserve the longest reliable rollouts possible, rather than to enforce short ones.

A.4. Bayesian and Partially Observable RL

Bayesian RL. Bayesian RL (Vlassis et al., 2012; Ghavamzadeh et al., 2015) models epistemic uncertainty (also known as ambiguity (Ellsberg, 1961)) for purposes such as exploration (Osband et al., 2016), robustness (Rajeswaran et al., 2017; Derman et al., 2020; Rigter et al., 2021), and generalization (Ghosh et al., 2021; Jiang et al., 2023). Uncertainty can be incorporated in model-free methods (e.g., Bayesian Q-learning (Dearden et al., 1998)) or in model-based methods (e.g., BAMDPs (Duff, 2002)). Depending on the design preference, Bayesian RL can be ambiguity-seeking, ambiguity-neutral, or ambiguity-averse. Our work is grounded in the epistemic POMDP formulation (Ghosh et al., 2021; 2022), an ambiguity-neutral, model-based framework for generalization through adaptation. This formulation stems from BAMDPs, which optimally balance the exploration-exploitation tradeoff but incur test-time exploratory costs, as illustrated in Sec. 3 and also known as the *cost of exploration* (Vuorio et al., 2024).

Epistemic POMDPs are also related to **meta-RL** (Beck et al., 2023), which likewise seeks to maximize expected performance over a distribution of MDPs. Each MDP is called a task in meta-RL. Bayesian meta-RL (Zintgraf et al., 2020; Dorfman et al., 2021) explicitly models this MDP posterior. The key difference is that in meta-RL the true environment is itself a pre-specified distribution over MDPs (a particular POMDP), so the task uncertainty is *inherent*. In contrast, the epistemic POMDP assumes a single underlying MDP, and its task uncertainty is purely epistemic. As a result, meta-RL methods cannot be applied to solve the epistemic POMDP without modification.

Recurrent model-free RL for online POMDPs. Model-free RL offers a simple and effective way to tackle online POMDPs without explicitly learning belief-state representations (Ni et al., 2022; Yang & Nguyen, 2021). Memory is typically implemented with recurrent neural networks (RNNs) such as LSTMs (Hochreiter & Schmidhuber, 1997) or

GRUs (Cho et al., 2014), but recent work shows that these architectures struggle with long-term memory (Parisotto et al., 2020; Ni et al., 2023). State-space models (SSMs) with linear recurrence (Orvieto et al., 2023; Gu & Dao, 2023) have emerged as a compute-efficient alternative to Transformers (Vaswani et al., 2017), balancing long-term memory with parallel optimization (Blelloch, 1990). Recent applications of SSMs to recurrent RL (Lu et al., 2023; 2024; Morad et al., 2024; Luo et al., 2024a; Luis et al., 2024) demonstrate strong performance on online POMDP benchmarks (Morad et al., 2023; Zintgraf et al., 2020; Ni et al., 2022).

A.5. Concurrent Work

ADM-v2 (Lin et al., 2026) is a concurrent work that, like ours, demonstrates the effectiveness of long-horizon rollouts for offline model-based RL. ADM-v2 pursues an extreme **full-horizon** setting, performing rollouts up to the maximum episode length (e.g., 1000 steps in D4RL locomotion tasks) and thus treating the learned world model as a surrogate simulator. Building on ADMPO (Lin et al., 2025), it simplifies multi-step modeling and introduces a parallelized rollout mechanism, enabling efficient any-step prediction. With full-horizon rollouts, ADM-v2 can perform off-policy evaluation directly without value bootstrapping and achieves strong performance on popular benchmarks when combined with an uncertainty penalty, representing the first successful full-horizon method.

We view ADM-v2 as highly complementary to our work. While ADM-v2 demonstrates that full-horizon rollouts can be effective under a *conservative* design, our method explores the opposite extreme: adhering to a Bayesian, *non-conservative* objective while using adaptive (but not full) horizons. Interestingly, ADM-v2's sensitivity analysis (Lin et al., 2026, Figure 16) shows that long horizons are most beneficial under strong penalties, whereas in *lightly penalized* regimes, excessively long horizons can degrade performance. This aligns with our findings: without explicit conservatism, performance is more sensitive to compounding errors, motivating adaptive truncation rather than full-horizon rollouts. Together, these two works suggest that long-horizon planning is a powerful component in offline RL, whose effective use depends on the degree of conservatism.

B. Extended Background on Conservative and Bayesian Principles

B.1. Formal Connections between Conservatism and Robustness

In this subsection, we place prior conservative algorithms in the (soft) robust MDP framework. For classic model-free and adversarial model-based pessimism, we make explicit how their updates correspond to particular uncertainty sets in *robust* MDPs (Wiesemann et al., 2013). For uncertainty-penalized pessimism, we connect it to *soft robust* MDP framework (Zhang et al., 2024b).[6] We follow the notation introduced in Sec. 2.

Classic model-free pessimism. Many model-free methods enforce an in-support (Fujimoto et al., 2019; Kumar et al., 2019) or in-sample (Kostrikov et al., 2022; Xu et al., 2023) constraint on the Bellman backup. This amounts to updating a pessimistic value function Q^{MF} such that, for a transition tuple $(s, a, r, d, s') \in \mathcal{D}$,

$$Q^{\mathrm{MF}}(s,a) \leftarrow r + \gamma(1-d) \max_{a' \in \mathcal{A},\, \text{s.t. } (s',a') \in \mathcal{D}} Q^{\mathrm{MF}}(s',a'). \tag{5}$$

This update is equivalent to assigning all out-of-dataset state–action pairs the minimal reward $-r_{\max}$, thereby enforcing a worst-case behavior. The corresponding uncertainty set $\mathfrak{M}_{\mathcal{D}}$ in the robust MDP framework (Eq. 1) can be written explicitly:

Proposition 1. *If the pessimistic update in Eq. 5 converges to a fixed point, then the induced uncertainty set for the corresponding robust MDP is*

$$\mathfrak{M}_{\mathcal{D}}^{\mathrm{MF}} = \left\{ m \;\middle|\; \begin{array}{ll} m(r, s' \mid s, a) = m_{\mathcal{D}}(r, s' \mid s, a), & \forall (s,a) \in \mathcal{D} \\ m(r, s' \mid s, a) = p(r)\, \mathbb{1}(s' = s_{absorb}), \;\; \forall p \in \Delta([-r_{\max}, r_{\max}]), & \forall (s,a) \notin \mathcal{D} \end{array} \right\}, \tag{6}$$

where $m_{\mathcal{D}}$ is the empirical model and $s_{absorb} \notin \mathcal{D}$ is an artificial absorbing state.

Proof. First, let $m \in \mathfrak{M}_{\mathcal{D}}^{\mathrm{MF}}$ and decompose $m(r, s' \mid s, a) = R(r \mid s, a) P(s' \mid s, a)$, and denote the empirical model as $m_{\mathcal{D}}(r, s' \mid s, a) = R_{\mathcal{D}}(r \mid s, a) P_{\mathcal{D}}(s' \mid s, a)$. By construction, $\mathfrak{M}_{\mathcal{D}}^{\mathrm{MF}}$ places no uncertainty on transitions: for $(s, a) \in \mathcal{D}$,

[6] The term "soft robustness" is used differently in prior work: Derman et al. (2018) use it for a Bayesian formalism, while Zhang et al. (2024b) define it as a risk-sensitive MDP relaxing strict worst-case robustness.

$P(s,a)$ equals the empirical transition $P_{\mathcal{D}}(s,a)$, while for $(s,a) \notin \mathcal{D}$, $P(s,a)$ deterministically transitions to s_{absorb}. Thus, all epistemic uncertainty is in the reward function R.

Applying the robust MDP framework (Wiesemann et al., 2013), the optimal value function Q^* is Markov and satisfies the robust Bellman optimality equation:

$$Q^*(s,a) = \min_{R(s,a) \in \mathfrak{M}_{\mathcal{D}}^{\text{MF}}(s,a)} \mathbb{E}_{r \sim R(s,a)}[r] + \gamma \mathbb{E}_{s' \sim P(s,a)} \left[\max_{a' \in \mathcal{A}} Q^*(s',a') \right], \forall (s,a) \in \mathcal{S} \times \mathcal{A}, \tag{7}$$

$$Q^*(s,a) - \gamma \mathbb{E}_{s' \sim P(s,a)} \left[\max_{a' \in \mathcal{A}} Q^*(s',a') \right] = \min_{R(s,a) \in \mathfrak{M}_{\mathcal{D}}^{\text{MF}}} \mathbb{E}_{r \sim R(s,a)}[r] = \begin{cases} \mathbb{E}_{r \sim R_{\mathcal{D}}(s,a)}[r] & (s,a) \in \mathcal{D}, \\ -r_{\max} & (s,a) \notin \mathcal{D}, \end{cases} \tag{8}$$

where the last line uses the fact that $R(\cdot \mid s,a)$ may be any distribution on $[-r_{\max}, r_{\max}]$, so the worst case is attained by a Dirac mass at $-r_{\max}$.

Therefore, we can simplify Eq. 8 by cases. (1) Absorbing state:

$$\forall a \in \mathcal{A}, \quad Q^*(s_{\text{absorb}}, a) - \gamma \max_{a' \in \mathcal{A}} Q^*(s_{\text{absorb}}, a') = -r_{\max}. \tag{9}$$

Since $Q^*(s_{\text{absorb}}, a)$ is a constant w.r.t. a, it follows that $Q^*(s_{\text{absorb}}, a) = -\frac{r_{\max}}{1-\gamma}, \forall a$, reaches the minimal return. (2) Unseen state-action pairs,

$$\forall (s,a) \notin \mathcal{D}, \quad Q^*(s,a) - \gamma \max_{a' \in \mathcal{A}} Q^*(s_{\text{absorb}}, a') = -r_{\max}. \tag{10}$$

This implies $Q^*(s,a) = -\frac{r_{\max}}{1-\gamma}, \forall (s,a) \notin \mathcal{D}$. (2) Seen state-action pairs,

$$\forall (s,a) \in \mathcal{D}, \quad Q^*(s,a) = \mathbb{E}_{r \sim R_{\mathcal{D}}(s,a)}[r] + \gamma \mathbb{E}_{s' \sim P_{\mathcal{D}}(s,a)} \left[\max_{a' \in \mathcal{A}} Q^*(s',a') \right] \tag{11}$$

$$= \mathbb{E}_{r \sim R_{\mathcal{D}}(s,a)}[r] + \gamma \mathbb{E}_{s' \sim P_{\mathcal{D}}(s,a)} \left[\max_{a' \in \mathcal{A}, \text{s.t. } (s',a') \in \mathcal{D}} Q^*(s',a') \right] \tag{12}$$

The last line follows that $Q^*(s, a_{\text{out}}) = -\frac{r_{\max}}{1-\gamma} \leq Q^*(s, a_{\text{in}}), \forall (s, a_{\text{in}}) \in \mathcal{D}, \forall (s, a_{\text{out}}) \notin \mathcal{D}$. Finally, Eq. 12 recovers the pessimism principle underlying Eq. 5. This includes many model-free offline RL algorithms, such as BCQ (Fujimoto et al., 2019, Equation 10), BEAR (Kumar et al., 2019, Definition 4.1), EMaQ (Ghasemipour et al., 2021, Theorem 3.3), IQL (Kostrikov et al., 2022, Corollary 2.1). □

Adversarial model-based pessimism. One class of model-based methods constructs an explicit uncertainty set around the empirical model (Uehara & Sun, 2022; Rigter et al., 2022):

$$\mathfrak{M}_{\mathcal{D}}^{\text{MB}} = \{m \mid \mathbb{E}_{(s,a) \sim \mathcal{D}}[\text{div}(m(s,a), m_{\mathcal{D}}(s,a))] \leq \epsilon\}, \tag{13}$$

where popular choices of the divergence $\text{div}(\cdot,\cdot)$ include total variation (TV) distance and KL divergence.

Uncertainty-penalized pessimism: soft robust MDP. Another line of model-based methods incorporates explicit uncertainty penalties into value updates (Yu et al., 2020; Kidambi et al., 2020; Jeong et al., 2023; Sun et al., 2023). For imagined transitions $(\hat{s}, \hat{a}, \hat{r}, \hat{d}, \hat{s}')$, the pessimistic update takes the form:

$$Q^{\text{MB}}(\hat{s}, \hat{a}) \leftarrow \hat{r} - \lambda U(\hat{s}, \hat{a}) + \gamma(1 - \hat{d}) \max_{\hat{a}' \in \mathcal{A}} Q^{\text{MB}}(\hat{s}', \hat{a}'), \tag{14}$$

where $U : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^+$ is an uncertainty measure based on dataset $\mathcal{D}$ and learned world models, and $\lambda > 0$ controls the degree of pessimism. Similar uncertainty penalties have also been incorporated into model-free value functions (Bai et al., 2022; An et al., 2021).

We now provide a connection between Eq. 14 and the *soft robust MDP* framework of (**?**)Section 5]zhang2024soft. While our derivation follows a similar line to theirs, we include it here to be self-contained. Consider a robust MDP with a *policy-dependent* uncertainty set:

$$\max_{\pi} \min_{m} J(\pi, m) \quad \text{s.t.} \quad \mathbb{E}_{(s,a) \sim (m,\pi)}[\text{div}(m(s,a), m^*(s,a))] \leq \epsilon, \tag{15}$$

where the divergence constraint describes the uncertainty set, taken in expectation under occupancy measure induced by m and π. We use Lagrangian relaxation with a coefficient $\alpha \geq 0$ to transform the problem into a soft robust MDP:

$$\max_{\pi} \min_{m} J(\pi, m) + \alpha \mathbb{E}_{(s,a)\sim(m,\pi)}[\mathrm{div}(m(s,a), m^*(s,a))]. \tag{16}$$

In dynamic programming form, for a given (s, a) pair, the inner optimization becomes

$$\min_{m(\cdot|s,a)} \mathbb{E}_{s'\sim m(\cdot|s,a)}\left[V^{\mathrm{MB}}(s')\right] + \alpha\, \mathrm{div}(m(\cdot \mid s,a), m^*(\cdot \mid s,a)), \tag{17}$$

where V^{MB} is the policy's state-value function. This inner problem can be transformed by duality, depending on the choice of divergence. For the KL divergence, one can apply Donsker and Varadhan's formula (Donsker & Varadhan, 1975)[7] to state that Eq. 17 is equivalent to

$$-\alpha \log\left(\mathbb{E}_{s'\sim m^*(\cdot|s,a)}\left[\exp\left(-\frac{V^{\mathrm{MB}}(s')}{\alpha}\right)\right]\right) = \mathbb{E}_{s'\sim m^*}\left[V^{\mathrm{MB}}(s')\right] - \frac{1}{2\alpha}\mathrm{Var}_{m^*}[V^{\mathrm{MB}}(s')] + O\left(\frac{1}{\alpha^3}\right), \tag{18}$$

where we use cumulant expansion.[8]

Therefore, the corresponding soft-robust Bellman optimality equation (Zhang et al., 2024b, Equation 15), ignoring higher-order terms, becomes

$$Q^{\mathrm{MB}}(s,a) = R^*(s,a) - \frac{\gamma}{2\alpha}\mathrm{Var}_{s'\sim P^*}[\max_{a'} Q^{\mathrm{MB}}(s',a')] + \gamma \mathbb{E}_{s'\sim P^*}\left[\max_{a'} Q^{\mathrm{MB}}(s',a')\right] \tag{19}$$

In practice, model-based RL methods approximate $m^* = (R^*, P^*)$ with an ensemble of learned models trained on $\mathcal{D}$. While the variance term in Eq. 19 reflects the aleatoric uncertainty of the true dynamics, ensemble-based variance also incorporates epistemic uncertainty by the law of total variance, thus blending both. Prior work has interpreted the penalty as aleatoric (Yu et al., 2020), epistemic (Sun et al., 2023), or both (Rigter et al., 2023); here we focus on its epistemic interpretation. Accordingly, the ensemble variance provides a practical surrogate for the penalty in Eq. 14.

B.2. Connection between Bayesianism and Partial Observability

Epistemic POMDP as a special class of POMDP. As noted by Ghosh et al. (2022, Appendix A), the Bayesian objective (Eq. 2) can be cast as a POMDP (Cassandra et al., 1994). We adapt their proof here. The POMDP's state space is $\mathcal{S}^+ = \mathcal{S} \times \mathfrak{M}_{\mathcal{D}}$ with the same action space $\mathcal{A}$, where $\mathfrak{M}_{\mathcal{D}} = \mathrm{supp}(\mathbb{P}_{\mathcal{D}})$. The joint reward–transition function in the POMDP is

$$\mathbb{P}(r_{t+1}, s^+_{t+1} \mid s^+_t, a_t) = \mathbb{1}(m_{t+1} = m_t) m_t(r_{t+1}, s_{t+1} \mid s_t, a_t),$$

where the initial state is $s^+_0 := (s_0, m_0)$ with $s_0 \sim \rho$ and $m_0 \sim \mathbb{P}_{\mathcal{D}}$. It is partially observable because the agent only observes $s_t \in s^+_t$, while the model $m_t \equiv m_0 \in \mathfrak{M}_{\mathcal{D}}$ remains hidden but fixed throughout each episode.

C. Analysis of Bootstrapped Error in Long-Horizon Rollouts

We formalize the insight in Sec. 4.1 using a result adapted from Sims et al. (2024, Proposition 1).

Consider a rollout generated by a deterministic world model m and a deterministic policy π,

$$\tau = (h_t, \hat{a}_t, \hat{r}_{t+1}, \hat{s}_{t+1}, \hat{a}_{t+1}, \ldots, \hat{s}_{t+H}),$$

where the initial history h_t is drawn from the dataset $\mathcal{D}$, $\hat{a}_{t+j} = \pi(\hat{h}_{t+j})$, $(\hat{r}_{t+j+1}, \hat{s}_{t+j+1}) = m(\hat{s}_{t+j}, \hat{a}_{t+j})$ and $\hat{h}_t = h_t$.

Assume a tabular policy evaluation setting and let Q^π denote the exact value function of policy π under the MDP m. Define the Bellman evaluation operator $\mathcal{T}$: for a value function Q,

$$\mathcal{T}Q(\hat{h}_{t+j}, \hat{a}_{t+j}) := \hat{r}_{t+j+1} + \gamma Q(\hat{h}_{t+j+1}, \pi(\hat{h}_{t+j+1})).$$

[7] For any probability distributions $x, x^* \in \Delta^k$ (the k-dimensional simplex), any vector $y \in \mathbb{R}^k$, and $\alpha > 0$, the duality formula is $-\alpha \log(\langle x^*, \exp(-y/\alpha)\rangle) = \min_x \langle x, y\rangle + \alpha\, \mathrm{KL}(x \,||\, x^*)$.

[8] For a random variable X, $\log(\mathbb{E}[\exp(tX)]) = t\mathbb{E}[X] + \frac{t^2}{2}\mathrm{Var}[X] + O(t^3)$. We substitute $t = -1/\alpha$.

We perform the value update along the imagined rollout backward in time. Let Q_0 be the initial value function prior to value update and Q_j be the value function at j-th iteration. For $j = 1, \dots, H$, define $Q_j(\hat{h}_{t+H-j}, \hat{a}_{t+H-j}) \approx \mathcal{T}Q_{j-1}(\hat{h}_{t+H-j}, \hat{a}_{t+H-j})$, with a per-step **TD error** bounded by:

$$|Q_j(\hat{h}_{t+H-j}, \hat{a}_{t+H-j}) - \mathcal{T}Q_{j-1}(\hat{h}_{t+H-j}, \hat{a}_{t+H-j})| \le \delta_j.$$

Following Sims et al. (2024), assume that $\hat{h}_{t+H}$ is an **edge-of-reach** history, i.e., a history used as Bellman targets but never itself updated. The **bootstrapped error** at this truncated point is

$$\epsilon := |Q_0(\hat{h}_{t+H}, \pi(\hat{h}_{t+H})) - Q^\pi(\hat{h}_{t+H}, \pi(\hat{h}_{t+H}))|.$$

We can decompose the total error for Q_j: for $j = 1, \dots, H$,

$$\begin{aligned}
\xi_j &:= |Q_j(\hat{h}_{t+H-j}, \hat{a}_{t+H-j}) - Q^\pi(\hat{h}_{t+H-j}, \hat{a}_{t+H-j})| \\
&\le |Q_j(\hat{h}_{t+H-j}, \hat{a}_{t+H-j}) - \mathcal{T}Q_{j-1}(\hat{h}_{t+H-j}, \hat{a}_{t+H-j})| \\
&+ |\mathcal{T}Q_{j-1}(\hat{h}_{t+H-j}, \hat{a}_{t+H-j}) - Q^\pi(\hat{h}_{t+H-j}, \hat{a}_{t+H-j})| \\
&\le \delta_j + |\gamma Q_{j-1}(\hat{h}_{t+H-j+1}, \hat{a}_{t+H-j+1}) - \gamma Q^\pi(\hat{h}_{t+H-j+1}, \hat{a}_{t+H-j+1})| \\
&= \delta_j + \gamma \xi_{j-1}.
\end{aligned} \tag{20}$$

Unrolling the recursion and using the fact that $\xi_0 = \epsilon$, the value at the final iteration is bounded by:

$$|Q_H(h_t, \hat{a}_t) - Q^\pi(h_t, \hat{a}_t)| = \xi_H \le \sum_{j=0}^{H-1} \gamma^j \delta_j + \gamma^H \epsilon. \tag{21}$$

The bound above is sign-agnostic, but under offline policy improvement the bootstrapped term at the truncation point tends to be *optimistic*. Near-greedy policies evaluate the critic on out-of-distribution actions that require extrapolation, where neural critics can assign spuriously high values. Even though edge-of-reach histories are not directly updated, their values can drift through function approximation and shared parameters, allowing optimistic errors to propagate. As a result, the terminal bootstrap often dominates value overestimation for short horizons, while increasing H mitigates this effect by exponentially down-weighting it.

D. Formal Existence Proof of the Advantage of Bayesianism

In this section, we compare the conservative and Bayesian principles in offline RL and highlight conditions under which the Bayesian approach yields provable advantages. After restating the conservative objective and its reliance on data coverage, we introduce a relaxed notion of coverage from a Bayesian view. This allows us to lower-bound the performance gap between Bayesian and robust solutions. We begin by defining the optimal policies in their respective objectives:

$$\pi^*(m^*) \in \operatorname*{argmax}_\pi J(\pi, m^*), \qquad \text{(ideal policy)}$$

$$\pi^*(\mathfrak{M}_\mathcal{D}) \in \operatorname*{argmax}_\pi J(\pi; \mathfrak{M}_\mathcal{D}), \qquad \text{(robust-optimal policy)}$$

$$\pi^*(\mathbb{P}_\mathcal{D}) \in \operatorname*{argmax}_\pi \mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}[J(\pi, m)]. \qquad \text{(Bayes-optimal policy)}$$

We then introduce the notion of *robust sub-optimality gap*, measuring the gap of the robust-optimal policy relative to the ideal one:

$$S_{\mathfrak{M}_\mathcal{D}}(m^*) := J(\pi^*(m^*), m^*) - J(\pi^*(\mathfrak{M}_\mathcal{D}), m^*). \tag{22}$$

To assess when $\pi^*(\mathfrak{M}_\mathcal{D})$ is competitive, one seeks an upper bound on this gap. Its tightness depends on dataset quality: the closer the data coverage is to $\pi^*(m^*)$, the smaller the gap. Prior work formalizes this dependence via coverage assumptions (Uehara & Sun, 2022; Li et al., 2024). We unify these notions through the following definition.

D.1. Comparison on Concentrability Coefficients

In this subsection, we define the robust and Bayesian concentrability coefficients and derive the numerical relationship between the two.

Definition 1 (Model-dependent concentrability)**.** *Given an MDP model m and a policy π, let d^π_m be the state-action occupancy measure of π on m, and let $\beta \in \Delta(\mathcal{S} \times \mathcal{A})$ be the offline distribution induced by the dataset $\mathcal{D}$ on m^*. The concentrability coefficient of π in MDP m under offline distribution β is defined as:*

$$\mathcal{C}(\pi, m) := \frac{\mathbb{E}_{(s,a)\sim d^\pi_m}\left[\mathrm{TV}(m(s,a), m^*(s,a))^2\right]}{\mathbb{E}_{(s,a)\sim\beta}[\mathrm{TV}(m(s,a), m^*(s,a))^2]}, \tag{23}$$

where TV *denotes the total variation distance between distributions.*

Intuitively, $\mathcal{C}(\pi, m)$ quantifies the mismatch between m and m^* under the policy distribution versus the dataset distribution.

We then extend the *model-based concentrability* of Uehara & Sun (2022, Definition 1), originally defined only for transition dynamics, to also incorporate the reward function. We refer to this generalization as robust concentrability.

Definition 2 (**Robust concentrability** (Uehara & Sun, 2022))**.** *Let $\mathfrak{M}_\mathcal{D}$ be a realizable hypothesis class of models consistently built from the dataset $\mathcal{D}$,*

$$\mathcal{C}(\pi) := \sup_{m\in\mathfrak{M}_\mathcal{D}} \mathcal{C}(\pi, m). \tag{24}$$

If $m^* \in \mathfrak{M}_\mathcal{D}$, the robust concentrability is upper bounded by the classic density-ratio-based concentrability, as shown in Uehara & Sun (2022, Section 4):

$$\mathcal{C}(\pi) \le \sup_{(s,a)\in\mathcal{S}\times\mathcal{A}} \frac{d^\pi_{m^*}(s,a)}{\beta(s,a)}.$$

We now consider an even *weaker* notion of coverage by extending model-dependent concentrability with a Bayesian posterior over models.

Definition 3 (**Bayesian concentrability**)**.**

$$\mathcal{C}_{\textit{Bayes}}(\pi) := \frac{\mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}\left[\mathbb{E}_{(s,a)\sim d^\pi_m}\left[\mathrm{TV}(m(s,a), m^*(s,a))^2\right]\right]}{\mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}\left[\mathbb{E}_{(s,a)\sim\beta}[\mathrm{TV}(m(s,a), m^*(s,a))^2]\right]}. \tag{25}$$

We now show that Bayesian concentrability is always upper bounded by its robust counterpart.

Proposition 2 (Bayesian concentrability is upper-bounded by robust concentrability)**.** *Assume that $\mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}[g(m)] > 0$, then*

$$C_{\textit{Bayes}}(\pi) \le \sup_{m\in\mathrm{supp}(\mathbb{P}_\mathcal{D})} \mathcal{C}(\pi, m).$$

Proof. Denote

$$f(m) := \mathbb{E}_{(s,a)\sim d^\pi_m}\left[\mathrm{TV}(m(s,a), m^*(s,a))^2\right],$$

$$g(m) := \mathbb{E}_{(s,a)\sim\beta}\left[\mathrm{TV}(m(s,a), m^*(s,a))^2\right],$$

so that

$$\mathcal{C}(\pi, m) = \frac{f(m)}{g(m)}, \quad \mathcal{C}_{\text{Bayes}}(\pi) := \frac{\mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}[f(m)]}{\mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}[g(m)]}.$$

$$\text{LHS} = \frac{\mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}\left[g(m)\frac{f(m)}{g(m)}\right]}{\mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}[g(m)]} \le \sup_{m\in\mathrm{supp}(\mathbb{P}_\mathcal{D})} \frac{f(m)}{g(m)} = \text{RHS},$$

by the elementary inequality $\sum_i w_i x_i \le \sum_i w_i \max_j x_j = (\sup_j x_j)\sum_i w_i$ for $w \ge 0$. □

The next example sharpens this bound to a strict inequality, making the Bayesian coverage assumption $\mathcal{C}_{\text{Bayes}}(\pi) < \infty$ strictly weaker than its robust analogue $\mathcal{C}(\pi) < \infty$.

Example 1 (Strictness of Bayesian concentrability bound)**.** *Let β have incomplete coverage over $\mathcal{S} \times \mathcal{A}$. Following the notations from the proof of Proposition 2, consider two models:*

1. *Model m_1 is a small perturbation of m^* on all of $\mathcal{S} \times \mathcal{A}$. Then, the numerator $f(m_1) < \infty$, the denominator $g(m_1) > 0$, making $\mathcal{C}(\pi, m_1) < \infty$.*
2. *Model m_2 is equivalent to m^* on $\text{supp}(\beta)$, i.e., $m_2(s,a) = m^*(s,a), \forall (s,a) \in \text{supp}(\beta)$, but there exists an off-support pair $(s^\dagger, a^\dagger) \notin \text{supp}(\beta)$ with $d^\pi_{m_2}(s^\dagger, a^\dagger) > 0$ and $\text{TV}(m_2(s^\dagger, a^\dagger), m^*(s^\dagger, a^\dagger)) > 0$. In that case, $g(m_2) = 0, f(m_2) > 0$, so $\mathcal{C}(\pi, m_2) = \infty$.*

If the posterior $\mathbb{P}_\mathcal{D}$ assigns weights $\mathbb{P}_\mathcal{D}(m_1) = 1 - \varepsilon$ and $\mathbb{P}_\mathcal{D}(m_2) = \varepsilon$ with any $\varepsilon \in (0,1)$, then

$$\mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}[f(m)] = (1-\varepsilon) f(m_1) + \varepsilon f(m_2) < \infty, \quad \mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}[g(m)] = (1-\varepsilon) g(m_1) > 0,$$

and $\mathcal{C}_{\text{Bayes}}(\pi) < \infty$, while $\sup_{m\in\text{supp}(\mathbb{P}_\mathcal{D})} \mathcal{C}(\pi, m) = \infty$.

D.2. Theorem 1 and Proof

To formalize the benefit of taking a Bayesian approach over a conservative one, we define the *Bayesian sub-optimality gap* as:

$$S_{\mathbb{P}_\mathcal{D}}(m^*) = J(\pi^*(m^*), m^*) - J(\pi^*(\mathbb{P}_\mathcal{D}), m^*), \tag{26}$$

which we aim to compare with the robust gap via their difference:

$$\Delta_{\mathbb{P}_\mathcal{D}, \mathfrak{M}_\mathcal{D}}(m^*) := S_{\mathfrak{M}_\mathcal{D}}(m^*) - S_{\mathbb{P}_\mathcal{D}}(m^*) = J(\pi^*(\mathbb{P}_\mathcal{D}), m^*) - J(\pi^*(\mathfrak{M}_\mathcal{D}), m^*). \tag{27}$$

The theorem below establishes a lower bound on this difference.

Theorem 1. *Assume that $\gamma > 1/2$. Then, we can construct a dataset $\mathcal{D}$, a set of MDPs $\mathfrak{M}_\mathcal{D}$, and a posterior $\mathbb{P}_\mathcal{D}$ such that $m^* \in \text{supp}(\mathbb{P}_\mathcal{D}) \subseteq \mathfrak{M}_\mathcal{D}$, and for some $\delta_0 > 0$, it holds with probability at least δ_0 that,*

$$\Delta_{\mathbb{P}_\mathcal{D}, \mathfrak{M}_\mathcal{D}}(m^*) > \frac{8 r_{\max}}{(1-\gamma)^2} \sqrt{\frac{\ln(|\mathfrak{M}_\mathcal{D}|/\delta_0)}{|\mathcal{D}|}} \left(\sqrt{\mathcal{C}(\pi^*(\mathfrak{M}_\mathcal{D}))} - \sqrt{\max(\mathcal{C}_{\textit{Bayes}}(\pi^*(m^*)), \mathcal{C}_{\textit{Bayes}}(\pi^*(\mathbb{P}_\mathcal{D})))} \right).$$

Proof sketch. The proof proceeds in two steps, each being established in Lemma 1 and Lemma 2 respectively. (1) We upper-bound the Bayesian gap $S_{\mathbb{P}_\mathcal{D}}(m^*)$ (Lemma 1). This bound makes explicit the "price of Bayesianism": coverage adequate for the ideal policy $\pi^*(m^*)$ alone does not suffice. We also need reasonable coverage for the Bayes-optimal policy, i.e., $\mathcal{C}_{\text{Bayes}}(\pi^*(\mathbb{P}_\mathcal{D})) < \infty$, even though posterior averaging makes $\mathcal{C}_{\text{Bayes}}(\cdot)$ more likely to be finite. (2) We lower-bound the robust gap $S_{\mathfrak{M}_\mathcal{D}}(m^*)$ (Lemma 2), by constructing an MDP with inadequate frequentist coverage relative to the behavior policy. □

Remark 1. This theorem shows that when $\mathcal{C}(\pi^*(\mathfrak{M}_\mathcal{D})) > \max(\mathcal{C}_{\text{Bayes}}(\pi^*(m^*)), \mathcal{C}_{\text{Bayes}}(\pi^*(\mathbb{P}_\mathcal{D})))$, the Bayes-optimal policy performs better than the robust-optimal policy in the true MDP. By construction, the lower bound in Lemma 2 holds when ϵ – the parametric gap between two well-chosen MDPs – is smaller than $c \cdot \mathcal{C}(\pi^*(\mathfrak{M}_\mathcal{D}))/|\mathcal{D}|$. This is more likely to occur when the dataset is small or when the concentrability $\mathcal{C}(\pi^*(\mathfrak{M}_\mathcal{D}))$ is large. In other words, adopting a frequentist (or conservative) approach rather than Bayesianism implies that the dataset does not provide sufficient statistical information when: (1) it is small, or (2) its coverage differs from that of the ground truth optimal policy. Those are the two features measuring the "quality" of an offline dataset. The Bayesian approach potentially yields better performance than conservatism because, for a given amount of data, the Bayesian concentrability is smaller than the robust one, making the lower bound irrelevant for the Bayesian setting.

Lemma 1 (Upper bound on Bayesian sub-optimality gap). *Assume that $m^* \in \text{supp}(\mathbb{P}_\mathcal{D}) \subseteq \mathfrak{M}_\mathcal{D}$. Then, for any $\delta \in (0,1)$, with probability at least $1-\delta$,*

$$S_{\mathbb{P}_\mathcal{D}}(m^*) \leq \frac{8 r_{\max}}{(1-\gamma)^2} \sqrt{\frac{\ln(|\mathfrak{M}_\mathcal{D}|/\delta)}{|\mathcal{D}|}} \sqrt{\max(\mathcal{C}_{\textit{Bayes}}(\pi^*(m^*)), \mathcal{C}_{\textit{Bayes}}(\pi^*(\mathbb{P}_\mathcal{D})))}.$$

Proof of Lemma 1. By definition of the Bayesian sub-optimality gap (Eq. 26),

$$\begin{aligned}
&S_{\mathbb{P}_\mathcal{D}}(m^*) \\
&= J(\pi^*(m^*), m^*) - J(\pi^*(\mathbb{P}_\mathcal{D}), m^*) \\
&= J(\pi^*(m^*), m^*) - \mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}[J(\pi^*(m^*), m)] + \mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}[J(\pi^*(m^*), m)] \\
&\quad - \mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}[J(\pi^*(\mathbb{P}_\mathcal{D}), m)] + \mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}[J(\pi^*(\mathbb{P}_\mathcal{D}), m)] - J(\pi^*(\mathbb{P}_\mathcal{D}), m^*) \\
&\leq (J(\pi^*(m^*), m^*) - \mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}[J(\pi^*(m^*), m)]) + (\mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}[J(\pi^*(\mathbb{P}_\mathcal{D}), m)] - J(\pi^*(\mathbb{P}_\mathcal{D}), m^*)),
\end{aligned}$$

where the last inequality holds by definition of $\pi^*(\mathbb{P}_\mathcal{D})$ being a solution of $\max_\pi \mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}[J(\pi, m)]$. The first term measures how the performance of the ideal policy $\pi^*(m^*)$ under the true MDP deviates from its Bayesian average. The second term measures the analogous deviation for the Bayes-optimal policy $\pi^*(\mathbb{P}_\mathcal{D})$. We aim to upper-bound each of these regret terms.

Step 1: Upper bound $J(\pi^*(m^*), m^*) - \mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}[J(\pi^*(m^*), m)]$.

$$\begin{aligned}
&J(\pi^*(m^*), m^*) - \mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}[J(\pi^*(m^*), m)] = \mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}[J(\pi^*(m^*), m^*) - J(\pi^*(m^*), m)] \\
&\leq \frac{2r_{\max}}{(1-\gamma)^2}\mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}\left[\mathbb{E}_{(s,a)\sim d_m^{\pi^*(m^*)}}[\mathrm{TV}(m(s,a), m^*(s,a))]\right],
\end{aligned}$$

according to Lemma 4. Therefore, by Jensen's inequality,

$$\begin{aligned}
J(\pi^*(m^*), m^*) - \mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}[J(\pi^*(m^*), m)] &\leq \frac{2r_{\max}}{(1-\gamma)^2}\mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}\left[\sqrt{\mathbb{E}_{(s,a)\sim d_m^{\pi^*(m^*)}}[\mathrm{TV}(m(s,a), m^*(s,a))^2]}\right] \\
&\leq \frac{2r_{\max}}{(1-\gamma)^2}\sqrt{\mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}\left[\mathbb{E}_{(s,a)\sim d_m^{\pi^*(m^*)}}[\mathrm{TV}(m(s,a), m^*(s,a))^2]\right]},
\end{aligned}$$

and by construction of the Bayesian concentrability coefficient (Eq. 25):

$$\begin{aligned}
&J(\pi^*(m^*), m^*) - \mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}[J(\pi^*(m^*), m)] \\
&\leq \frac{2r_{\max}}{(1-\gamma)^2}\sqrt{\mathcal{C}_{\text{Bayes}}(\pi^*(m^*))\mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}\left[\mathbb{E}_{(s,a)\sim\beta}[\mathrm{TV}(m(s,a), m^*(s,a))^2]\right]}.
\end{aligned}$$

Finally, since by construction $\mathbb{P}_\mathcal{D}$ is supported only on models that achieve MLE under $\mathcal{D}$, every $m \in \mathrm{supp}(\mathbb{P}_\mathcal{D})$ satisfies the PAC bound of Lemma 3, yielding

$$J(\pi^*(m^*), m^*) - \mathbb{E}_{m\sim\mathbb{P}_\mathcal{D}}[J(\pi^*(m^*), m)] \leq \frac{4r_{\max}}{(1-\gamma)^2}\sqrt{\mathcal{C}_{\text{Bayes}}(\pi^*(m^*))}\sqrt{\frac{\ln(|\mathfrak{M}_\mathcal{D}|/\delta)}{|\mathcal{D}|}}.$$

Step 2: Upper bound $\mathbb{E}_{m\sim\mathbb{P}}[J(\pi^*(\mathbb{P}_\mathcal{D}), m)] - J(\pi^*(\mathbb{P}_\mathcal{D}), m^*)$. We use the same reasoning as above, but applied this time to the Bayes-optimal policy, and deduce the following bound:

$$\mathbb{E}_{m\sim\mathbb{P}}[J(\pi^*(\mathbb{P}_\mathcal{D}), m)] - J(\pi^*(\mathbb{P}_\mathcal{D}), m^*) \leq \frac{4r_{\max}}{(1-\gamma)^2}\sqrt{\mathcal{C}_{\text{Bayes}}(\pi^*(\mathbb{P}_\mathcal{D}))}\sqrt{\frac{\ln(|\mathfrak{M}_\mathcal{D}|/\delta)}{|\mathcal{D}|}}.$$

We finally obtain:

$$\begin{aligned}
S_{\mathbb{P}_\mathcal{D}}(m^*) &\leq \frac{4r_{\max}}{(1-\gamma)^2}\sqrt{\frac{\ln(|\mathfrak{M}_\mathcal{D}|/\delta)}{|\mathcal{D}|}}\left(\sqrt{\mathcal{C}_{\text{Bayes}}(\pi^*(m^*))} + \sqrt{\mathcal{C}_{\text{Bayes}}(\pi^*(\mathbb{P}_\mathcal{D}))}\right) \\
&\leq \frac{8r_{\max}}{(1-\gamma)^2}\sqrt{\frac{\ln(|\mathfrak{M}_\mathcal{D}|/\delta)}{|\mathcal{D}|}}\sqrt{\max(\mathcal{C}_{\text{Bayes}}(\pi^*(m^*)), \mathcal{C}_{\text{Bayes}}(\pi^*(\mathbb{P}_\mathcal{D}))}.
\end{aligned}$$

□

Lemma 2 (Existence of a lower bound on robust sub-optimality gap). *Assume that $\gamma > 1/2$. We can construct an MDP instance $m^* = m_1$ inducing an offline dataset $\mathcal{D}$ and a set of models $\mathfrak{M} = \{m_{-1}, m_1\}$, such that the optimal robust policy $\pi^*(\mathfrak{M})$ satisfies w.p. at least δ_0:*

$$J(\pi^*(m^*), m^*) - J(\pi^*(\mathfrak{M}), m^*) > \frac{8r_{\max}}{(1-\gamma)^2}\sqrt{\frac{\ln(|\mathfrak{M}|/\delta_0)}{|\mathcal{D}|}}\sqrt{\mathcal{C}(\pi^*(\mathfrak{M}))}.$$

Proof of Lemma 2. Similarly as (Li et al., 2024), we build two MDPs (here, two sequential two-armed bandits) and a behavior policy such that the conservative value estimated from the resulting dataset is far from the optimal return.

The example. Consider two sequential bandits $\mathfrak{M} := \{m_{-1}, m_1\}$ such that $m^* = m_1$ is the ground-truth model. We parameterize both by $\theta \in \{-1, 1\}$. They share the same action space $\mathcal{A} := \{-1, 1\}$, the same reward for the negative action, but different reward for the positive action, namely, $R_\theta(-1) \sim \mathcal{B}(1/2)$ while $R_\theta(1) \sim \mathcal{B}(1/2 + \theta\epsilon)$ with $\epsilon > 0$ that will be determined later. Clearly, the ideal policy for each MDP is $\pi^*_\theta = \theta$. With notational abuse, let the behavior policy be $\beta(1) = \beta = 1 - \beta(-1)$ from which we collect $|\mathcal{D}|$ i.i.d. samples. As will be justified in the following, we set $\beta = \frac{(1-\gamma)^4}{64 r_{\max}^2 \mathcal{C}(\pi^*(\mathfrak{M}))} \in [0, 1]$.

Relation to the bandit example in Sec. 3. Here, action -1 plays the role of the sampling action with Bernoulli parameter $1/2$, and action 1, the "uncovered action" in the skewed bandit dataset of Sec. 3. We deliberately avoid setting $\beta(1) = 0$, since that would violate the data coverage assumption and make $\mathcal{C}(\pi^*(\mathfrak{M}))$ infinite.[9] Yet, the dataset is still skewed because it can be generated from m_1 under a suboptimal policy (but close-to-optimal policy for m_{-1}). In that case, it is statistically hard to identify the correct model *when ϵ is too small*, which is how we choose it for the lower bound to hold.

Proof. With a slight abuse of notation, denote a policy by $\pi := \pi(1) = 1 - \pi(-1)$. The discounted value may be expressed according to the underlying MDP as:

$$J(\pi, \theta) = \sum_{t=0}^{\infty} \gamma^t \left(\pi \cdot (1/2 + \theta\epsilon) + (1 - \pi) \cdot 1/2)\right) = \sum_{t=0}^{\infty} \gamma^t \left(\frac{1}{2} + \theta\epsilon\pi\right) = \frac{1 + 2\theta\epsilon\pi}{2(1-\gamma)}.$$

For each model $\theta \in \{-1, 1\}$, the suboptimality gap of a policy π is thus:

$$\delta^\pi(1) = \frac{1 + 2\epsilon}{2(1-\gamma)} - \frac{1 + 2\epsilon\pi}{2(1-\gamma)} = \frac{2\epsilon(1-\pi)}{2(1-\gamma)} = \frac{\epsilon(1-\pi)}{1-\gamma},$$
$$\delta^\pi(-1) = \frac{1}{2(1-\gamma)} - \frac{1 - 2\epsilon\pi}{2(1-\gamma)} = \frac{\epsilon\pi}{1-\gamma}.$$

More synthetically, and since $\gamma \in [0, 1)$, we get:

$$\delta^\pi(\theta) = \frac{\epsilon}{1-\gamma}(1 - \pi(\theta)). \tag{28}$$

By contradiction, suppose that we can find a policy estimate from the dataset such that:

$$\mathbb{P}_\theta(J(\pi^*(\theta), \theta) - J(\hat{\pi}, \theta) \leq \epsilon) = \mathbb{P}_\theta(\delta^{\hat{\pi}}(\theta) \leq \epsilon) \geq \frac{7}{8}.$$

Then, from Eq. 28, we should have with probability greater than $7/8$ that:

$$\frac{\epsilon}{1-\gamma}(1 - \hat{\pi}(\theta)) \leq \epsilon \iff \hat{\pi}(\theta)\frac{\epsilon}{1-\gamma} \geq \frac{\epsilon}{1-\gamma} - \epsilon \iff \hat{\pi}(\theta) \geq 1 - (1-\gamma) = \gamma.$$

By assumption, $\gamma > 1/2$. If the above statement were true, then we could construct the following estimator $\hat{\theta}$ of θ based on $\hat{\pi}$:

$$\hat{\theta} = \operatorname*{argmax}_{a \in \{-1,1\}} \hat{\pi}(a) \tag{29}$$

and thus, $\mathbb{P}_\theta(\hat{\theta} = \theta) = \mathbb{P}_\theta(\hat{\pi}(\theta) > 1/2) \geq \mathbb{P}_\theta(\hat{\pi}(\theta) \geq \gamma) \geq 7/8$.

We analyze the hypothesis testing of identifying the true MDP given the generated data. Formally, let the test $\phi(\mathcal{D}) = 0$ mean "decide $m^* = m_{-1}$" and $\phi(\mathcal{D}) = 1$ mean "decide $m^* = m_1$". Consider the minimax probability of error p_e between m_{-1} and m_1:

$$p_e \;=\; \inf_{\phi} \max(\mathbb{P}_{-1}(\phi(\mathcal{D}) \neq 0),\; \mathbb{P}_1(\phi(\mathcal{D}) \neq 1)),$$

[9] In the limit $\mathcal{C}(\pi^*(\mathfrak{M})) \to \infty$, we recover the example in Sec. 3.

where $\mathbb{P}_\theta(\mathcal{D})$ denotes the sampling distribution of $\mathcal{D}$ under model θ. By (Bickel et al., 2009)[Thm. 2.2], the following lower bound holds:

$$p_e \geq \frac{1}{4}\exp(-\mathrm{KL}(\mathbb{P}_{-1}(\mathcal{D}) \,||\, \mathbb{P}_1(\mathcal{D}))). \tag{30}$$

Each data sample $\mathcal{D}_i$ is generated according to a mixture of two Bernoullis $\mathcal{D}_i \sim \beta\mathcal{B}(1/2+\theta\epsilon) + (1-\beta)\mathcal{B}(1/2)$. Let n_w be the number of 1-reward samples, i.e., successful events $\mathcal{D}_i = 1$ (respectively, n_l the number of 0-reward samples). Then:

$$\begin{aligned}
\mathbb{P}(\mathcal{D}_i = 1) &= \beta\left(\frac{1}{2}+\theta\epsilon\right) + (1-\beta)\frac{1}{2} = \frac{1}{2}+\beta\theta\epsilon, \\
\mathbb{P}(\mathcal{D}_i = 0) &= \beta\left(1-\left(\frac{1}{2}+\theta\epsilon\right)\right) + (1-\beta)\left(1-\frac{1}{2}\right) = \beta\left(\frac{1}{2}-\theta\epsilon\right) + \frac{1}{2} - \frac{\beta}{2} = \frac{1}{2}-\beta\theta\epsilon,
\end{aligned}$$

and the likelihood for the whole dataset is:

$$\mathbb{P}_\theta(\mathcal{D}) = \left(\frac{1}{2}+\beta\theta\epsilon\right)^{n_\mathrm{w}}\left(\frac{1}{2}-\beta\theta\epsilon\right)^{n_\mathrm{l}}. \tag{31}$$

Based on Eq. (31), we can compute the divergence:

$$\mathrm{KL}(\mathbb{P}_1(\mathcal{D}) \,\|\, \mathbb{P}_{-1}(\mathcal{D})) = n_\mathrm{w}\mathrm{KL}\left(\mathcal{B}\left(\frac{1}{2}+\beta\epsilon\right), \mathcal{B}\left(\frac{1}{2}-\beta\epsilon\right)\right) + n_\mathrm{l}\mathrm{KL}\left(\mathcal{B}\left(\frac{1}{2}-\beta\epsilon\right), \mathcal{B}\left(\frac{1}{2}+\beta\epsilon\right)\right),$$

by the additive property of KL between independent samples. Remarking that

$$\begin{aligned}
&\mathrm{KL}\left(\mathcal{B}\left(\frac{1}{2}+\epsilon\beta\right), \mathcal{B}\left(\frac{1}{2}-\epsilon\beta\right)\right) \\
&= \left(\frac{1}{2}+\epsilon\beta\right)\log\left(\frac{\frac{1}{2}+\epsilon\beta}{\frac{1}{2}-\epsilon\beta}\right) + \left(1-\frac{1}{2}-\epsilon\beta\right)\log\left(\frac{1-\frac{1}{2}-\epsilon\beta}{1-\frac{1}{2}+\epsilon\beta}\right) \\
&= \left(\frac{1}{2}+\epsilon\beta\right)\log\left(\frac{\frac{1}{2}+\epsilon\beta}{\frac{1}{2}-\epsilon\beta}\right) + \left(\frac{1}{2}-\epsilon\beta\right)\log\left(\frac{\frac{1}{2}-\epsilon\beta}{\frac{1}{2}+\epsilon\beta}\right) \\
&= \mathrm{KL}\left(\mathcal{B}\left(\frac{1}{2}-\epsilon\beta\right), \mathcal{B}\left(\frac{1}{2}+\epsilon\beta\right)\right),
\end{aligned}$$

it results that

$$\begin{aligned}
\mathrm{KL}(\mathbb{P}_{-1}(\mathcal{D}) \,\|\, \mathbb{P}_1(\mathcal{D})) &= (n_\mathrm{w}+n_\mathrm{l})\mathrm{KL}\left(\mathcal{B}\left(\frac{1}{2}+\epsilon\beta\right), \mathcal{B}\left(\frac{1}{2}-\epsilon\beta\right)\right) \\
&= |\mathcal{D}|\left(\frac{1}{2}+\epsilon\beta\right)\log\left(\frac{\frac{1}{2}+\epsilon\beta}{\frac{1}{2}-\epsilon\beta}\right) + |\mathcal{D}|\left(\frac{1}{2}-\epsilon\beta\right)\log\left(\frac{\frac{1}{2}-\epsilon\beta}{\frac{1}{2}+\epsilon\beta}\right) \\
&= |\mathcal{D}|\left(\frac{1}{2}+\epsilon\beta-\frac{1}{2}+\epsilon\beta\right)\log\left(\frac{\frac{1}{2}+\epsilon\beta}{\frac{1}{2}-\epsilon\beta}\right) \\
&= 2|\mathcal{D}|\epsilon\beta\log\left(\frac{1+2\epsilon\beta}{1-2\epsilon\beta}\right).
\end{aligned}$$

For a small enough ϵ, the series expansion of the logarithm term yields:

$$\log\left(\frac{1+2\epsilon\beta}{1-2\epsilon\beta}\right) = 2\cdot 2\epsilon\beta + o((2\epsilon\beta)^2) = 4\epsilon\beta + o((\epsilon\beta)^2)$$

so that

$$\mathrm{KL}(\mathbb{P}_{-1}(\mathcal{D}) \,\|\, \mathbb{P}_1(\mathcal{D})) = 2|\mathcal{D}|\epsilon\beta(4\epsilon\beta + o((\epsilon\beta)^2)) = 8|\mathcal{D}|(\epsilon\beta)^2 + o((\epsilon\beta)^3).$$

We can eventually deduce that for a small enough ϵ:

$$\mathrm{KL}(\mathbb{P}_{-1}(\mathcal{D}) \,\|\, \mathbb{P}_1(\mathcal{D})) \leq c|\mathcal{D}|\epsilon^2\beta. \tag{32}$$

The binary testing lower bound (30) together with Eq. 32 establishes that the misidentification probability p_e is at least $1/8$ as long as

$$\exp\left(-c|\mathcal{D}|\epsilon^2\beta\right) \geq 1/2, \tag{33}$$

which is equivalent to $\epsilon \leq \sqrt{\frac{\ln(2)}{c|\mathcal{D}|\beta}}$.

To conclude, we have assumed that there is a policy estimate $\hat{\pi}$ such that $\mathbb{P}_\theta(\delta^{\hat{\pi}}(\theta) \leq \epsilon) \geq 7/8$, namely, $\mathbb{P}_{-1}(\delta^{\hat{\pi}}(-1) > \epsilon) < 1/8$ and $\mathbb{P}_1(\delta^{\hat{\pi}}(1) > \epsilon) < 1/8$. Then, in view of our previous arguments, the estimator $\hat{\theta}$ as defined in Eq. 29 must satisfy $\mathbb{P}_{-1}(\hat{\theta} \neq \theta) < 1/8$ and $\mathbb{P}_{-1}(\hat{\theta} \neq \theta) < 1/8$, which contradicts the misidentification lower-bound $p_e \geq 1/8$ when $\epsilon \leq \sqrt{\frac{\ln(2)}{c|\mathcal{D}|\beta}}$. Remarking that in our example, $|\mathfrak{M}| = 2$ and $\delta_0 = 1/8$, we set $c = \ln(2)/\ln(16)$ to conclude that any policy estimate $\hat{\pi}$ inevitably satisfies $\mathbb{P}_\theta(J(\pi^*(\theta), \theta) - J(\hat{\pi}, \theta) \geq \epsilon) \geq \frac{7}{8}$[10]. In particular, the statement holds for the optimal robust policy $\pi^*(\mathfrak{M})$. □

D.3. Auxiliary Lemmas for Theorem 1

We first recall a standard PAC bound for maximum likelihood estimation (MLE), adapted from Agarwal et al. (2020a, Theorem 21).

Lemma 3 (MLE PAC-bound). *Let $\beta \in \Delta(\mathcal{S} \times \mathcal{A})$ be the offline distribution induced by $\mathcal{D}$, and*

$$\hat{m} = \operatorname*{argmax}_{m \in \mathfrak{M}} \mathbb{E}_{(s,a,r,s') \sim \mathcal{D}}[\log m(r, s' \mid s, a)]$$

be the MLE model within a finite uncertainty set $\mathfrak{M}$. Suppose $m^* \in \mathfrak{M}$. Then, for any $\delta \in (0, 1)$, with probability at least $1 - \delta$:

$$\mathbb{E}_{(s,a)\sim\beta}\left[\mathrm{TV}(\hat{m}(s,a), m^*(s,a))^2\right] \leq \frac{2\log(|\mathfrak{M}|/\delta)}{|\mathcal{D}|}.$$

We next establish a *general* simulation lemma below. Unlike the classic simulation lemma (Uehara & Sun, 2022, Lemma 9), which applies only to stationary policies and considers transition errors alone, our result (i) extends to history-dependent policies via general Bellman recursions, and (ii) incorporates discrepancies in both transition and reward functions.

Lemma 4 (General simulation lemma). *For any history-dependent policy $\pi : \mathcal{H}_t \to \Delta(\mathcal{A})$ and any two MDP models $m = (P, R), \hat{m} = (\hat{P}, \hat{R})$, it holds that:*

$$|J(\pi, m) - J(\pi, \hat{m})| \leq \frac{2r_{\max}}{(1-\gamma)^2}\mathbb{E}_{(s,a)\sim d^\pi_{\hat{m}}}[\mathrm{TV}(m(s,a), \hat{m}(s,a))].$$

Proof. Given an MDP m and a policy π, define the value function starting from a history $h_t \in \mathcal{H}_t$:

$$V^\pi_m(h_t) := \mathbb{E}_{\pi,m}\left[\sum_{k=t}^{\infty} \gamma^{k-t} r_{k+1} | h_t\right].$$

It satisfies the history-based Bellman recursion, where $h_{t+1} = (h_t, a_t, r_{t+1}, s_{t+1})$,

$$V^\pi_m(h_t) = \mathbb{E}_{a_t\sim\pi(h_t)}\left[\mathbb{E}_{r_{t+1}\sim R(s_t,a_t), s_{t+1}\sim P(s_t,a_t)}[r_{t+1} + \gamma V^\pi_m(h_{t+1})]\right]$$

[10] For that specific value $c = \ln(2)/\ln(16)$, any $\epsilon < \min\{\frac{1}{\beta}\sqrt{1/4 - 8\beta}), 1/(2\beta)\}$ would make the inequality (33) valid, as long as $\beta < 1/32$. Since $\mathcal{C}(\pi^*(\mathfrak{M})) \geq 1$ and $r_{\max} = 1$, condition $\beta < 1/32$ is automatically fulfilled. Additionally, for $\beta < 1/32$, it holds that $\sqrt{\frac{\ln(2)}{c|\mathcal{D}|\beta}} \leq \frac{1}{\beta}\sqrt{1/4 - 8\beta}$, so $\epsilon \leq \sqrt{\frac{\ln(2)}{c|\mathcal{D}|\beta}}$ is a more restrictive bound on ϵ.

$$= \mathbb{E}_{a_t\sim\pi(h_t)}\big[\mathbb{E}_{(r_{t+1},s_{t+1})\sim m(s_t,a_t)}[r_{t+1}+\gamma V_m^\pi(h_{t+1})]\big]. \tag{34}$$

Define the value difference between models m and $\hat{m}$ given a history $h_t \in \mathcal{H}_t$:

$$\Delta^\pi(h_t) := V_m^\pi(h_t) - V_{\hat{m}}^\pi(h_t).$$

Thus the return gap is

$$|J(\pi, m) - J(\pi, \hat{m})| = |\mathbb{E}_{s_0\sim\rho}[\Delta^\pi(s_0)]|.$$

Applying recursion (34) to both models m and $\hat{m}$, we get:

$$\Delta^\pi(h_t) = \mathbb{E}_{a_t\sim\pi(h_t)}\big[\mathbb{E}_{m(s_t,a_t)}[r_{t+1}+\gamma V_m^\pi(h_{t+1})] - \mathbb{E}_{\hat{m}(s_t,a_t)}[r_{t+1}+\gamma V_{\hat{m}}^\pi(h_{t+1})]\big].$$

Decompose this into two parts:

$$\begin{aligned}
\Delta_1^\pi(h_t,a_t) &:= \mathbb{E}_{m(s_t,a_t)}[r_{t+1}+\gamma V_m^\pi(h_{t+1})] - \mathbb{E}_{\hat{m}(s_t,a_t)}[r_{t+1}+\gamma V_m^\pi(h_{t+1})],\\
\Delta_2^\pi(h_t,a_t) &:= \mathbb{E}_{\hat{m}(s_t,a_t)}[r_{t+1}+\gamma V_m^\pi(h_{t+1})] - \mathbb{E}_{\hat{m}(s_t,a_t)}[r_{t+1}+\gamma V_{\hat{m}}^\pi(h_{t+1})],\\
\Delta^\pi(h_t) &= \mathbb{E}_{a_t\sim\pi(h_t)}[\Delta_1^\pi(h_t,a_t)+\Delta_2^\pi(h_t,a_t)].
\end{aligned} \tag{35}$$

Applying the fundamental property of TV distance (Levin & Peres, 2017), the first term:

$$\begin{aligned}
\Delta_1^\pi(h_t,a_t) &\le 2\mathrm{TV}(m(s_t,a_t),\hat{m}(s_t,a_t))\cdot\|r_{t+1}+\gamma V_m^\pi(h_{t+1})\|_\infty\\
&\le 2\mathrm{TV}(m(s_t,a_t),\hat{m}(s_t,a_t))\frac{r_{\max}}{1-\gamma},
\end{aligned} \tag{36}$$

where the second inequality stems from the fact that:

$$\|r_{t+1}+\gamma V_m^\pi(h_{t+1})\|_\infty \le r_{\max} + \gamma\frac{r_{\max}}{1-\gamma} = \frac{r_{\max}}{1-\gamma}.$$

The second term can be recognized as:

$$\Delta_2(h_t,a_t) = \gamma\mathbb{E}_{(r_{t+1},s_{t+1})\sim\hat{m}(s_t,a_t)}[\Delta^\pi(h_{t+1})]. \tag{37}$$

Combining Eq. 35 with Equations (36) and (37) yields:

$$\Delta^\pi(h_t) \le \mathbb{E}_{a_t\sim\pi(h_t)}\left[\frac{2r_{\max}}{1-\gamma}\mathrm{TV}(m(s_t,a_t),\hat{m}(s_t,a_t))\right] + \gamma\mathbb{E}_{a_t\sim\pi(h_t)}\big[\mathbb{E}_{\hat{m}(s_t,a_t)}[\Delta^\pi(h_{t+1})]\big].$$

For convenience, denote the one-step error difference at time t by:

$$\epsilon(s_t,a_t) := \frac{2r_{\max}}{1-\gamma}\mathrm{TV}(m(s_t,a_t),\hat{m}(s_t,a_t)),$$

so that

$$\Delta^\pi(h_t) \le \mathbb{E}_{a_t\sim\pi(h_t)}[\epsilon(s_t,a_t)] + \gamma\mathbb{E}_{a_t\sim\pi(h_t)}\big[\mathbb{E}_{(r_{t+1},s_{t+1})\sim\hat{m}(s_t,a_t)}[\Delta^\pi(h_{t+1})]\big].$$

Iterating from $t=0$ and taking $\mathbb{E}_{s_0\sim\rho}$,

$$\begin{aligned}
&\mathbb{E}_{s_0\sim\rho}[\Delta^\pi(s_0)]\\
&\le \mathbb{E}_{s_0\sim\rho,a_0\sim\pi(s_0)}[\epsilon(s_0,a_0)] + \gamma\mathbb{E}_{s_0\sim\rho,a_0\sim\pi(s_0),(r_1,s_1)\sim\hat{m}(s_0,a_0)}[\Delta^\pi(h_1)]\\
&= \mathbb{E}_{h_0\sim P^\pi_{\hat{m},0},a_0\sim\pi(h_0)}[\epsilon(s_0,a_0)] + \gamma\mathbb{E}_{h_1\sim P^\pi_{\hat{m},1}}[\Delta^\pi(h_1)]\\
&\le \mathbb{E}_{h_0\sim P^\pi_{\hat{m},0},a_0\sim\pi(h_0)}[\epsilon(s_0,a_0)] + \gamma\left(\mathbb{E}_{h_1\sim P^\pi_{\hat{m},1},a_1\sim\pi(h_1)}[\epsilon(s_1,a_1)] + \gamma\mathbb{E}_{h_2\sim P^\pi_{\hat{m},2}}[\Delta^\pi(h_2)]\right)\\
&= \mathbb{E}_{h_0\sim P^\pi_{\hat{m},0},a_0\sim\pi(h_0)}[\epsilon(s_0,a_0)] + \gamma\mathbb{E}_{h_1\sim P^\pi_{\hat{m},1},a_1\sim\pi(h_1)}[\epsilon(s_1,a_1)] + \gamma^2\mathbb{E}_{h_2\sim P^\pi_{\hat{m},2}}[\Delta^\pi(h_2)]\\
&\vdots
\end{aligned}$$

$$\leq \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{h_t \sim P^{\pi}_{\hat{m},t}, a_t \sim \pi(h_t)}[\epsilon(s_t, a_t)]$$
$$= \frac{1}{1-\gamma} \mathbb{E}_{(s,a)\sim d^{\pi}_{\hat{m}}}[\epsilon(s,a)],$$

where in the last line, we use the discounted occupancy $d^{\pi}_{\hat{m}}$ under $\hat{m}$:

$$d^{\pi}_{\hat{m}}(s,a) = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \mathbb{P}(s_t = s, a_t = a \mid \pi, \hat{m}).$$

Finally, by symmetry (the same argument applied to $-\mathbb{E}_{s_0\sim\rho}[\Delta^{\pi}(s_0)]$),

$$\begin{aligned}|\mathbb{E}_{s_0\sim\rho}[\Delta^{\pi}(s_0)]| &\leq \left|\frac{1}{1-\gamma}\mathbb{E}_{(s,a)\sim d^{\pi}_{\hat{m}}}[\epsilon(s,a)]\right| \\ &= \frac{1}{1-\gamma}\mathbb{E}_{(s,a)\sim d^{\pi}_{\hat{m}}}\left[\frac{2r_{\max}}{1-\gamma}\mathrm{TV}(m(s,a),\hat{m}(s,a))\right] \\ &= \frac{2r_{\max}}{(1-\gamma)^2}\mathbb{E}_{(s,a)\sim d^{\pi}_{\hat{m}}}[\mathrm{TV}(m(s,a),\hat{m}(s,a))].\end{aligned}$$

□

D.4. Related Work on Offline RL Theory

Theoretical works on offline RL aim to establish performance guarantees without online exploration. The key difficulty is limited data coverage: when the dataset sufficiently covers all state-action pairs, standard PAC-style guarantees can be obtained without requiring conservatism (Munos & Szepesvári, 2008). Under partial coverage, however, conservative algorithms that penalize policies deviating from well-supported regions are essential to ensure robust learning. Many prior results establish minimax optimality with information-theoretic bounds under *worst-case* assumptions over possible MDPs (Jin et al., 2021; Rashidinejad et al., 2021; Li et al., 2024). The lower-bound part in Lemma 2 draws on the same information-theoretic principle as Li et al. (2024), but we provide a simpler counterexample tailored to our sequential bandit problem in Sec. 3.

Complementary to this line of work, our work adopts a Bayesian viewpoint that focuses on the *average case* over possible MDPs, an aspect underexplored in the theoretical literature. An exception is the Bayesian offline RL setting studied in Uehara & Sun (2022, Section 8) under a Markov policy, optimized via mirror descent with posterior sampling. Their analysis derives a Bayesian sub-optimality gap and interprets the resulting soft robustness as "implicit pessimism". However, because their analysis is restricted to Markov policies, whereas the Bayes-optimal policy is generally history-dependent, their regret bound is looser than ours in Lemma 1.

Recent work (Fellows et al., 2025) addresses the same Bayesian offline RL problem with history-dependent policies. Their main result (Theorem 1) provides a Bayesian sub-optimality bound expressed by an expected KL divergence, called the posterior information loss, and takes the worst case over all policies. Their derivation starts from a TV distance, which they upper bound by a KL divergence to leverage the product rule of logarithms. In that respect, our Lemma 1 gives a tighter upper bound as it is based on TV distance rather than KL, which requires extending the simulation lemma to history-dependent policies (see Lemma 4). Additionally, the regret bound provided in Fellows et al. (2025, Theorem 1) involves a supremum over all policies. Although we believe their bound can be established directly on the Bayes-optimal policy rather than the worst case, building on their current result would require a full data-coverage assumption. In contrast, our lower bound in Lemma 1 only requires partial data coverage. Finally, Fellows et al. (2025) focuses on bounding the Bayes regret. At the same time, we additionally provide a lower bound on the robust sub-optimality gap in Lemma 2, showing when the Bayesian approach can provably outperform conservative ones.

E. NEUBAY Algorithm Details

RL loss function. In recurrent off-policy RL, optimization is typically performed on full trajectories (rather than i.i.d. transition tuples) for compute efficiency. In our setting, each training trajectory is a *concatenation* of a real prefix and an

imagined rollout: starting from an initial history $h_t \in \mathcal{D}$, a world model and the policy generate future steps until truncation at $t' \leq T$. Formally, let

$$\tau = (s_{0:t} \oplus s_{t+1:t'}, a_{0:t-1} \oplus a_{t:t'-1}, r_{1:t} \oplus \hat{r}_{t+1:t'}, d_{1:t} \oplus \hat{d}_{t+1:t'}),$$

where $\oplus$ denotes *concatenation*. Our RL loss on τ balances contributions from real and imagined segments:

$$L(Q_\omega, \pi_\nu; \tau, \kappa) := \frac{\kappa}{t} \sum_{j=0}^{t-1} l(Q_\omega, \pi_\nu; h_j, a_j, r_{j+1}, d_{j+1}, s_{j+1}) + \frac{1-\kappa}{t'-t} \sum_{j=t}^{t'-1} l(Q_\omega, \pi_\nu; \hat{h}_j, \hat{a}_j, \hat{r}_{j+1}, \hat{d}_{j+1}, \hat{s}_{j+1}). \tag{38}$$

Here, $\hat{h}_j$ denotes the imagined history (with $\hat{h}_t = h_t$) and $\kappa \in (0, 1)$ is the real data ratio. The per-step loss $l(Q_\omega, \pi_\nu; h_j, a_j, r_{j+1}, d_{j+1}, s_{j+1})$ is standard off-policy loss without explicit conservatism, such as DQN (Mnih et al., 2013) for discrete control and SAC (Haarnoja et al., 2018a) for continuous control.

Rollout stopping criteria. In our rollout subroutine (Algo. 2), a rollout finishes when any of three conditions hold:

$$\text{done}_{t+1} := (U_{\boldsymbol{\theta}}(\hat{s}_t, \hat{a}_t) > \mathcal{U}(\zeta)) \vee (t+1 \geq T) \vee f_{\text{term}}(\hat{s}_t, \hat{a}_t, \hat{s}_{t+1}). \tag{39}$$

1. Uncertainty truncation: as described in Sec. 4.2, instead of enforcing a fixed horizon H, we truncate rollouts adaptively using an uncertainty threshold calibrated on the real dataset. This allows rollouts to extend as long as the model remains confident.
2. Timout truncation: to remain consistent with test-time evaluation, we impose a hard cap at the environment's maximum episode length T, regardless of rollout length.
3. Ground-truth termination: we retain the environment's rule-based terminal function to provide true terminal signals $\hat{d}_{t+1}$, following prior model-based RL methods (Yu et al., 2020). Including this prior knowledge makes our algorithm directly comparable to model-based baselines, which are our main focus.

Importantly, only the terminal signal disables bootstrapping in RL, while both uncertainty- and timeout-based truncations preserve bootstrapping[11], which aligns with the Bayesian objective.

F. Implementation Details

F.1. Reproducibility Statement

We release our full codebase in the supplementary material, built on JAX (Bradbury et al., 2018) and Equinox (Kidger & Garcia, 2021). For further reproducibility, we also release pretrained world ensemble checkpoints to make agent training on top of the released checkpoints straightforward.

F.2. Dataset Details

The bandit dataset. As introduced in Sec. 3, we construct a skewed two-armed bandit dataset. We collect 10 trajectories of length $T = 100$, yielding $|\mathcal{D}| = 1000$ action–reward pairs. We use one-hot encoding on actions as inputs for reward models and agents. The dataset is split into training and validation sets with a 4:1 ratio. Since $\mathcal{D}$ only covers arm 0 with $p_0^* = 0.5$, the true parameter of arm 1, p_1^*, is completely unseen.

At test time, we vary $p_1^* \in \{0.01, 0.3, 0.55, 0.7, 0.99\}$, where each choice defines a distinct bandit problem. Each problem is evaluated over 20 independent episodes, and all problems are assessed in parallel, yielding $5 \times 20 = 100$ evaluation runs. The normalized return during test time is computed by: $\frac{1}{T} \sum_{t=0}^{T-1} r_{t+1}$, where $r_{t+1} \sim \mathcal{B}(p_{a_t}^*)$.

D4RL locomotion benchmark. We evaluate on the standard D4RL locomotion benchmark (Fu et al., 2020), comprising 12 datasets formed by the Cartesian product of tasks (halfcheetah, hopper, walker2d) and dataset types (random-v2, medium-v2, medium-replay-v2, medium-expert-v2). This benchmark is the most widely used in offline RL research. The underlying environments are OpenAI Gym tasks: HalfCheetah-v2 ($\mathcal{S} \subset \mathbb{R}^{17}, \mathcal{A} \subset \mathbb{R}^6$), Hopper-v2 ($\mathcal{S} \subset \mathbb{R}^{11}, \mathcal{A} \subset \mathbb{R}^3$), and Walker2d-v2 ($\mathcal{S} \subset \mathbb{R}^{17}, \mathcal{A} \subset \mathbb{R}^6$). The maximum episode step T is 1000. Hopper-v2 and Walker2d-v2 have termination functions, while HalfCheetah-v2 does not.

Dataset sizes vary: 100k-200k transitions for medium-replay, 1M for random and medium, and 2M for medium-expert. These datasets also differ qualitatively. The random dataset is collected with a uniformly random policy; medium-replay

[11] https://gymnasium.farama.org/tutorials/gymnasium_basics/handling_time_limits/.

corresponds to the replay buffer of an agent trained to a medium-level policy; medium itself is generated directly from a medium-level policy; and medium-expert is a mixture of trajectories from both a medium policy and an expert policy. Based on these properties, we categorize random as *low-quality*, medium-replay and medium-expert as *moderate coverage*, and medium as *narrow coverage*.

Performance is reported in terms of normalized scores, following the D4RL and NeoRL conventions:

$$\text{normalized score} = \frac{\text{score} - \text{random score}}{\text{expert score} - \text{random score}} \times 100.$$

It is worth noting that the expert policies in these locomotion tasks are not strictly optimal. As a result, it is possible for algorithms to achieve normalized scores greater than 100 (e.g., around 120). Therefore, we consistently categorize all medium-$\star$ datasets as medium-quality.

NeoRL locomotion benchmark. The locomotion benchmark in the NeoRL (Qin et al., 2022) has been widely used in recent model-based offline RL methods. The setup closely mirrors the D4RL locomotion benchmark, with nearly identical environments: HalfCheetah-v3 ($\mathcal{S} \subset \mathbb{R}^{18}, \mathcal{A} \subset \mathbb{R}^{6}$), Hopper-v3 ($\mathcal{S} \subset \mathbb{R}^{12}, \mathcal{A} \subset \mathbb{R}^{3}$), and Walker2d-v3 ($\mathcal{S} \subset \mathbb{R}^{18}, \mathcal{A} \subset \mathbb{R}^{6}$). Compared to D4RL, NeoRL increases the state dimensionality by one in each environment.

For each environment, NeoRL provides three datasets (Low, Medium, High), collected using policies of the corresponding performance levels. This leads to 9 datasets in total. Compared to their D4RL counterparts, these policies are more deterministic, leading to smaller data coverage, which we categorize as *narrow coverage* in this paper. Dataset sizes range from roughly 200k to 1M transitions. To avoid ambiguity, throughout the paper we denote NeoRL datasets with capitalized environment names and the v3 suffix (e.g., *HalfCheetah-v3-Medium*), while D4RL datasets are written in lowercase with the v2 suffix (e.g., *halfcheetah-medium-v2*).

D4RL Adroit benchmark. The Adroit benchmark is widely regarded as substantially more challenging than locomotion tasks. It involves controlling a 28-DoF robotic arm to perform high-dimensional manipulation tasks: Pen ($\mathcal{S} \subset \mathbb{R}^{45}, \mathcal{A} \subset \mathbb{R}^{24}, T = 100$), Door ($\mathcal{S} \subset \mathbb{R}^{39}, \mathcal{A} \subset \mathbb{R}^{28}, T = 200$), Hammer ($\mathcal{S} \subset \mathbb{R}^{46}, \mathcal{A} \subset \mathbb{R}^{26}, T = 200$). Among these, Pen includes a termination function, while Door and Hammer do not. Rewards combine a dense component based on distances with a sparse component that grants a large bonus once a distance threshold is satisfied.

In addition to the high dimensionality and sparsity of rewards, most Adroit datasets are either small or of limited quality. The human demonstration datasets (human-v1) contain only 5k–10k transitions, whereas the cloned datasets (cloned-v1), constructed by mixing behavior cloning with human demonstrations, contain 500k–1M transitions. This leads to 6 datasets in total. Accordingly, we categorize human datasets as *narrow coverage* and cloned datasets as *moderate coverage*.

Dataset performance levels vary significantly, as shown in the $\pi_{\mathcal{D}}$ column of Tab. 11. Pen-human-v1 and pen-cloned-v1 achieve normalized scores in the range of 70–90, whereas door-human-v1, door-cloned-v1, hammer-human-v1, and hammer-cloned-v1 yield scores below 10. Accordingly, we categorize pen-$\star$ datasets as *medium-quality*, and the remaining ones as *low-quality*. This disparity explains why most algorithms fail to achieve meaningful results on the door and hammer datasets. Finally, we exclude the relocate tasks from our experiments, as both prior baselines and our method consistently obtain near-zero scores in this setting.

D4RL AntMaze benchmark. AntMaze is a particularly challenging benchmark in D4RL due to its sparse reward structure. It involves controlling a MuJoCo Ant ($\mathcal{S} \subset \mathbb{R}^{29}, \mathcal{A} \subset \mathbb{R}^{8}$) to navigate in different maze layouts (umaze, medium, large). The agent receives reward 1 only when reaching a goal position in the X–Y plane within a threshold, after which the episode terminates, so the maximum return is 1. Goals are randomly sampled from a small region but remain unobserved to the agent, making evaluation noisy for RL algorithms. The maximum episode length is $T = 700$ for umaze and $T = 1000$ for medium and large mazes.

For each layout, two datasets (play and diverse) are provided, each containing 1M transitions. The *diverse* variants introduce a wider set of start positions compared to *play*. However, all AntMaze datasets are generated by a hierarchical controller: a high-level breadth-first search (BFS) planner selects waypoints, which are then executed by a low-level learned policy. As a result, trajectories are highly structured and largely restricted to narrow corridors that connect goals without collisions. This planner-driven generation induces narrow coverage, a property noted in prior work (Zhang et al., 2024a).

Performance in AntMaze is generally low relative to other D4RL domains. In particular, umaze-diverse has an average score of only 1.0, which we categorize as *low-quality*. The remaining AntMaze datasets achieve scores above 10.0, which

we categorize as *medium-quality* (though they could also be viewed as low-quality when considered outside the AntMaze domain).

Following prior model-based RL work (LEQ (Park & Lee, 2025)[12], ADMPO (Lin et al., 2025)[13]), we adopt the same terminal functions in AntMaze. As in LEQ, we also shift the reward by -1.0. Since termination reveals information about the reward function (termination implies success), we do not compare against other model-based or model-free algorithms. Instead, our baselines in AntMaze are the methods reported in LEQ and ADMPO.

F.3. Details on World Model Ensemble

We provide implementation details on world ensemble, as introduced in Sec. 2 and Sec. 4.3.

Ensemble architecture. Following common practice in offline RL (Yu et al., 2020), our world ensemble is a set of neural networks $\mathbf{m}_{\boldsymbol{\theta}} = \{m_{\theta^n}\}_{n=1}^{N}$ where each model outputs a Gaussian distribution over next state s' and reward r: $m_{\theta^n}(s,a) = \mathcal{N}(\mu_{\theta^n}(s,a), \sigma_{\theta^n}(s,a))$, with parameters $\boldsymbol{\theta} = \{\theta^n\}_{n=1}^{N}$ independently initialized at random. Following MOBILE (Sun et al., 2023; Sun, 2023)[14], each MLP $m_\theta(s,a)$ has 4 hidden layers with a width of 200 for the locomotion benchmarks, and 5 layers with a width of 400 for Adroit benchmark. As described in Sec. 4.3, we apply LayerNorm after each hidden layer; this design choice is further ablated in Sec. 5.2. The basic block consists of (Linear $\to$ LayerNorm $\to$ leaky ReLU) when LayerNorm is enabled, and (Linear $\to$ leaky ReLU) when it is disabled. For the bandit task, we use a small reward-model ensemble that has 2 hidden layers with a width of 16. The weights of each MLP are initialized independently using the default `equinox.nn.Linear` scheme, i.e., $\mathrm{Unif}[-\frac{1}{\sqrt{\dim_{\text{in}}}}, \frac{1}{\sqrt{\dim_{\text{in}}}}]$, where $\dim_{\text{in}}$ denotes the input feature dimension.

Ensemble selection. For each dataset, we train an initial pool of 128 MLPs and select the top N models to form the world model ensemble $\mathbf{m}_{\boldsymbol{\theta}}$, which remains fixed during subsequent policy training. Model selection is based on a held-out validation set. For continuous control tasks, we rank the models by MSE on a validation set consisting of 1000 transitions from $\mathcal{D}$, following MOPO (Yu et al., 2020). For the bandit task, we rank by negative log-likelihood (NLL) that captures uncertainty, since the true reward follows Bernoulli distribution. In our main experiments (Sec. 5.1), we use $N = 100$. For the sensitivity analysis (Sec. 5.2), we vary $N \in \{5, 20\}$.

Ensemble training. Each model is trained using standard maximum likelihood estimation (MLE). For continuous control tasks, we sample transition tuples (s, a, r, s') from $\mathcal{D}$ using a batch size of 256 to estimate the MLE loss. The inputs (s, a) and outputs (r, s') are standardized by subtracting the mean and dividing by the standard deviation computed over the dataset; during inference, predictions are inverse-transformed to restore the original scale. We use AdamW optimizer (Loshchilov & Hutter, 2019) with a weight decay coefficient of 5×10^{-5}, and a learning rate of 1×10^{-3} for locomotion tasks and 3×10^{-4} for Adroit tasks. Training is terminated early if the validation MSE fails to improve by more than 0.01 relative within five consecutive epochs, following the early stopping procedure in MOBILE. In the bandit task, the model learning rate is 1×10^{-3}, batch size is 128, and improvement threshold is 0.001 absolute.

F.4. Details on Planning

At the start of planning, following Sec. 4.2, we sample a batch of initial histories $h_t \sim \mathcal{D}$, drawn uniformly across time steps. The batch size is fixed at 100, regardless of the ensemble size N. Since the values of N used in our experiments $\{5, 20, 100\}$ are divisors of 100, we distribute the histories evenly across ensemble members. The recurrent policy π_ν initializes its hidden state z_t with h_t and then interacts with each world model to generate corresponding rollout until reaching the stopping criterion described in Sec. E. The uncertainty threshold $\mathcal{U}(\zeta)$ for truncation is fixed at $\zeta = 1.0$ in our main experiments and varied to $\{0.9, 0.99, 0.999\}$ in the sensitivity analysis. The entire planning process is parallelized on a GPU and thus introduces only negligible time costs.

For the bandit task, since there are no transition dynamics and hence no compounding error, we directly optimize the Bayesian objective: planning starts at $t = 0$ and truncates at $t = T$.

[12] https://github.com/kwanyoungpark/LEQ.
[13] https://github.com/HxLyn3/ADMPO.
[14] https://github.com/yihaosun1124/mobile.

F.5. Details on Recurrent Off-Policy RL

Off-policy loss implementation. For continuous control tasks, we follow a recent recurrent off-policy RL algorithm RESeL (Luo et al., 2024a) to use REDQ (Chen et al., 2021b) as the per-step loss for $l(\cdot)$, used in the overall RL loss defined by Eq. 38. REDQ builds on SAC (Haarnoja et al., 2018a;b)[15], maintaining an ensemble of 10 critic MLPs and sampling 2 of them to form bootstrapped targets in the critic loss, while the actor maximizes the average Q-value over all ensemble members. REDQ uses in-target minimization to reduce overestimation, and may induce mild underestimation. In our use of REDQ, it *mainly* serves as a critic-stabilization component, because our notion of "explicit conservatism" refers to the deliberate injection of pessimism into offline policy learning. See Sec. G.4 for an ablation replacing REDQ with standard SAC to show that REDQ plays a secondary role on NEUBAY.

For the bandit task, we adopt dueling DQN (Wang et al., 2016) following memoroid (Morad et al., 2024) as the discrete control algorithm. Exploration is ϵ-greedy, annealed from 1.0 to 0.1 over the first 10% of gradient steps.

Agent architecture implementation. As introduced in Sec. 4.4, our agent consists of a recurrent actor $\pi_\nu : \mathcal{H}_t \to \Delta(\mathcal{A})$ and a recurrent critic $Q_\omega : \mathcal{H}_t \times \mathcal{A} \to \mathbb{R}^{10}$. The critic outputs an ensemble of 10 Q-values, following the REDQ design adopted in RESeL. Both actor and critic maintain their own RNN encoders, $\nu_\phi : \mathcal{H}_t \to \mathcal{Z}$ and $\omega_\phi : \mathcal{H}_t \to \mathcal{Z}$, which share the same architecture but are optimized independently. As mentioned earlier, we adopt the *memoroid* framework (Morad et al., 2024)[16] to use the linear recurrent unit (LRU) (Orvieto et al., 2023) as the backbone encoder for both ν_ϕ and ω_ϕ. The actor and critic MLP heads have 2 or 3 hidden layers with a hidden size of 256. For the bandit task, there is only a critic MLP head with 2 hidden layers.

The LRU begins with a nonlinear preprocessing layer that projects the raw input history h_t into a 256-dimensional feature space (preserving the time dimension). This is followed by a stack of two LRU layers, each performing a linear recurrence update parameterized by a complex-valued diagonal matrix. Each layer maintains a hidden state of size 128, resulting in a recurrent representation $z_t \in \mathbb{C}^{2\times 128}$. From this recurrent state, the model produces a real-valued output vector $\tilde{z}_t \in \mathbb{R}^{128}$ via a nonlinear projection. During training and inference, $\tilde{z}_t$ is fed into the actor or critic MLP heads, while the complex hidden state z_t is preserved for recurrent updates during policy inference.

We fix the recurrent architecture, including hidden sizes, across all experiments. However, NEUBAY is compatible with any RNN encoder, and we leave a study of architectural variations to future work. For the ablation study with Markov agent, we remove the LRU encoder and only train the actor-critic MLP.

Tape-based batching. Classic recurrent RL relies on *segment-based batching* (Ni et al., 2022), where sequences are padded to a fixed length, forming 3D tensors of shape $(\text{batch size}, \text{sequence length}, \text{dim})$ with NaN masks for shorter sequences. This wastes memory and lowers sample efficiency. Memoroid (Morad et al., 2024) introduces *tape-based batching*, which exploits the monoid algebra of linear RNNs such as LRUs. Instead of padding, variable-length sequences are concatenated into a single 2D "tape", making the effective batch size equal to the sum of raw sequence lengths. Inline resets of hidden states prevent leakage across sequences within a tape (Lu et al., 2023), and computation over the tape is parallelized using associative scan in JAX (Bradbury et al., 2018). To enable JIT compilation, a fixed tape length is enforced by dropping any trailing timesteps that exceed this length. We refer readers to Morad et al. (2024) for full details.

In our implementation, we adopt tape-based batching and we set the tape length to be larger than the maximal episode length T in each task. To reduce computation time, we choose a relatively small tape length, similar to prior work in online POMDPs (Morad et al., 2024; Luo et al., 2024a).

Training hyperparameters. Tab. 3 summarizes the modules and hyperparameters fixed in our experiments. Consistent with IQL (Kostrikov et al., 2022) and MOBILE (Sun et al., 2023), we apply cosine learning rate decay only to the actor network (both the RNN encoder and MLP head), but not to the critic, in order to promote stability during the later stages of training.

The REDQ (SAC) entropy coefficient α is auto-tuned with a target entropy of $-\dim(\mathcal{A})$ (Haarnoja et al., 2018b) for all datasets except for D4RL and NeoRL Hopper datasets and D4RL AntMaze domain. In the D4RL hopper domain, we find our algorithm is sensitive to α, consistent with prior recurrent RL work (Luo et al., 2024a). To address this, we follow MOBILE and fix $\alpha = 0.2$ across all hopper datasets (four in D4RL and three in NeoRL), without further tuning.

In AntMaze, sparse-reward navigation makes SAC sensitive to the choice of α, e.g., ADMPO (Lin et al., 2025) uses a fixed

[15] Our REDQ implementation is adapted from the SAC-N Equinox codebase: `https://github.com/Howuhh/sac-n-jax`.

[16] `https://github.com/proroklab/memoroids`.

$\alpha = 0.05$ and disables the entropy term backup in critic loss. We follow their insights but instead tune the target entropy $\in \{-\dim(\mathcal{A}), -5\dim(\mathcal{A}), -10\dim(\mathcal{A})\}$. We find $-10\dim(\mathcal{A})$ works best overall and report it in our main results.

For the bandit task, we sweep the RNN encoder learning rate $\eta_\phi \in \{1 \times 10^{-6}, 3 \times 10^{-6}, 1 \times 10^{-5}, 3 \times 10^{-5}, 1 \times 10^{-4}\}$ in the critic network. Consistent with observations from ReSEL (Luo et al., 2024a) and our continuous control results, a small learning rate is crucial for stable training. We use $\eta_\phi = 3 \times 10^{-6}$ in our bandit experiments.

Table 3. Fixed hyperparameters used in our recurrent agents. The last block (actor and policy entropy) is only used in continuous control.

Module or Hyperparameter	Value
Actor and critic RNN encoders	2-layer LRUs (Orvieto et al., 2023)
RNN hidden state size	256
Actor and critic heads	2-layer MLPs (3 layers in Adroit)
Basic block of MLP head	(Linear $\to$ LayerNorm $\to$ leaky ReLU)
MLP head hidden size	256
Batch size (i.e., tape length)	2048 (1024 in Adroit, 1000 in bandit)
Update-to-data (UTD) ratio	0.05 (0.02 in bandit)
Gradient steps	2M (3M in AntMaze, 20k in bandit)
Replay buffer size	Full size, i.e., (60M in AntMaze, 1M in bandit, 40M otherwise)
Discount factor γ	0.99
Critic head's learning rate	1×10^{-4}
Gradient norm clipping	1000 (10000 in Adroit, 1 in bandit and AntMaze)
Actor head's learning rate	1×10^{-4}
Actor's learning rate decay	Cosine decay to 0.0
Entropy coef. α's learning rate	1×10^{-4}
Entropy coef. α	Auto-tuned with target $-\dim(\mathcal{A})$ (AntMaze uses $-10\dim(\mathcal{A})$ while Hopper uses fixed $\alpha = 0.2$)

Table 4. Best hyperparameters per dataset in the D4RL locomotion benchmark. We sweep $\eta_\phi \in \{3 \times 10^{-7}, 1 \times 10^{-6}, 3 \times 10^{-6}, 1 \times 10^{-5}, 3 \times 10^{-5}\}$ and $\kappa \in \{0.05, 0.5, 0.8\}$.

Dataset	RNN encoder lr η_ϕ	Real data ratio κ
halfcheetah-random-v2	3×10^{-5}	0.8
hopper-random-v2	1×10^{-5}	0.5
walker2d-random-v2	3×10^{-5}	0.5
halfcheetah-medium-replay-v2	1×10^{-5}	0.05
hopper-medium-replay-v2	3×10^{-7}	0.5
walker2d-medium-replay-v2	1×10^{-6}	0.5
halfcheetah-medium-v2	3×10^{-5}	0.8
hopper-medium-v2	1×10^{-6}	0.8
walker2d-medium-v2	3×10^{-6}	0.5
halfcheetah-medium-expert-v2	3×10^{-5}	0.8
hopper-medium-expert-v2	3×10^{-6}	0.5
walker2d-medium-expert-v2	1×10^{-6}	0.8

Table 5. Best hyperparameters per dataset in the NeoRL locomotion benchmark. We sweep $\eta_\phi \in \{3 \times 10^{-7}, 1 \times 10^{-6}, 3 \times 10^{-6}, 1 \times 10^{-5}, 3 \times 10^{-5}\}$ and $\kappa \in \{0.5, 0.8\}$.

Dataset	RNN encoder lr η_ϕ	Real data ratio κ
HalfCheetah-v3-Low	1×10^{-5}	0.8
Hopper-v3-Low	3×10^{-6}	0.8
Walker2d-v3-Low	3×10^{-7}	0.5
HalfCheetah-v3-Medium	3×10^{-5}	0.5
Hopper-v3-Medium	3×10^{-6}	0.5
Walker2d-v3-Medium	3×10^{-7}	0.8
HalfCheetah-v3-High	3×10^{-5}	0.5
Hopper-v3-High	3×10^{-6}	0.5
Walker2d-v3-High	3×10^{-7}	0.8

Table 6. Best hyperparameters per dataset in the D4RL Adroit benchmark. We sweep $\eta_\phi \in \{1 \times 10^{-6}, 3 \times 10^{-6}, 1 \times 10^{-5}, 3 \times 10^{-5}, 1 \times 10^{-4}\}$ and $\kappa \in \{0.5, 0.8\}$. For the remaining Adroit datasets, all hyperparameter settings yield near-zero performance.

Dataset	RNN encoder lr η_ϕ	Real data ratio κ
pen-human-v1	3×10^{-5}	0.5
pen-cloned-v1	1×10^{-5}	0.8
hammer-cloned-v1	1×10^{-5}	0.5

Table 7. Best hyperparameters per dataset in the D4RL AntMaze benchmark. We sweep $\eta_\phi \in \{1 \times 10^{-6}, 3 \times 10^{-6}, 1 \times 10^{-5}, 3 \times 10^{-5}\}$ and $\kappa \in \{0.5, 0.8, 0.95\}$. For the large maze datasets, all hyperparameter settings yield near-zero performance.

Dataset	RNN encoder lr η_ϕ	Real data ratio κ
antmaze-umaze-v2	3×10^{-6}	0.95
antmaze-umaze-diverse-v2	3×10^{-6}	0.95
antmaze-medium-play-v2	1×10^{-5}	0.95
antmaze-medium-diverse-v2	1×10^{-6}	0.95

F.6. Computation Details

This subsection provides additional details on rollout costs, model training time, and parameter size to complement the summary in Sec. 5.3.

Table 8. **Time cost (seconds) of the Rollout function** (Algo. 2) for different ensemble sizes N and numbers of parallel rollouts K. In our main experiments, $N = K = 100$.

	$K = 5$	$K = 20$	$K = 100$
$N = 5$	2.7s	3.0s	4.7s
$N = 20$	N/A	3.6s	5.0s
$N = 100$	N/A	N/A	5.3s

Rollout costs. Tab. 8 reports the time cost of our Rollout implementation for halfcheetah-medium-expert-v2. We assign each ensemble member an equal number of rollouts, so we only benchmark configurations where K is divisible by N. Thanks to full vectorization over ensemble members and rollouts using `jax.vmap`, rollout inference is very efficient: increasing N or K has only a minor effect on runtime. Consequently, rollout cost is negligible, and agent training time is dominated by gradient updates rather than rollout generation.

Table 9. **World model training time (per seed)** for different total ensemble size N_{total}. In practice, we use $N_{\text{total}} = 128$.

$N_{\text{total}} = 8$	$N_{\text{total}} = 32$	$N_{\text{total}} = 128$
1.2 hrs (1x)	1.7 hrs (1.42x)	6.0 hrs (5x)

Model training time. For completeness, Tab. 9 reports the training time for several choices of N_{total} on halfcheetah-medium-expert-v2 to provide additional compute context. The training time is *roughly sublinear* in N_{total}. In practice, we train one ensemble of size $N_{\text{total}} = 128$ and select the top $N = 5, 20, 100$.

Parameter size. In the locomotion benchmarks, the world model ensemble with $N = 100$ contains fewer than 15M parameters in total. The recurrent encoder has fewer than 600k parameters, the actor MLP head fewer than 200k, and the critic MLP head (with an ensemble size of 10) fewer than 1.5M.

G. Further Results and Discussion

G.1. Full Benchmarking Results

Table 10. Comparison of offline RL methods on the **NeoRL locomotion** benchmark. We report mean normalized scores for all baselines, with ±std for competitive baselines. The **best mean score** is bolded, and marked methods are statistically similar under a t-test. Our results use 6 seeds, each evaluated at the final step with 20 episodes.

	Model-free		Conservative model-based								Bayesian-inspired		Ours
Dataset	EDAC	CQL	MOPO	COMBO	ROMI	MOBILE	LEQ	ADMPO	ScorePen	VIPO	MAPLE	MoDAP	NEUBAY
hc-Low	31.3	38.2	40.1	32.9	35.8±3.9	54.7±3.0	33.4±1.6	52.8±1.2	49.6±1.2	**58.5**±0.1	33.4	53.9±1.1	53.1±1.1
hp-Low	18.3	16.0	6.2	17.9	22.4±0.5	17.4±3.9	24.2±2.3	22.3±0.1	21.1±2.3	**30.7**±0.3	22.7	26.1±4.7	30.3±2.9
wk-Low	40.2	44.7	11.6	31.7	36.4±2.3	37.6±2.0	65.1±2.3	55.9±3.8	51.4±1.4	**67.6**±0.7	33.9	51.3±7.8	43.9±5.9
hc-Med	54.9	54.6	62.3	50.8	57.7±1.4	77.8±1.4	59.2±3.9	69.3±1.7	77.4±1.0	80.9±0.2	69.5	81.0±2.3	**81.1**±0.8
hp-Med	44.9	64.5	1.0	56.3	46.6±6.8	51.1±13.3	**104.3**±5.2	51.5±5.0	90.9±1.3	66.3±0.2	27.7	44.2±15.3	95.7±11.5
wk-Med	57.6	57.3	39.9	53.8	54.9±2.0	62.2±1.6	45.2±19.4	70.1±2.4	65.8±1.6	**76.8**±0.1	40.7	70.8±3.1	50.5±8.8
hc-High	81.4	77.4	65.9	62.2	77.4±2.7	83.0±4.6	71.8±8.0	84.0±0.8	81.4±1.0	**89.4**±0.6	—	84.1±8.3	68.3±23.6
hp-High	52.5	76.6	11.5	63.2	65.9±4.5	87.8±26.0	95.5±13.9	87.6±4.9	86.3±1.3	**107.7**±0.5	—	52.4±3.2	96.8±7.7
wk-High	75.5	75.3	18.0	71.8	75.1±1.8	74.9±3.4	73.7±1.1	**82.2**±1.9	78.0±1.8	81.7±1.0	—	73.6±2.8	62.7±14.5
AVG	50.7	56.1	28.5	49.0	52.5	60.7	63.6	64.0	66.9	**73.3**	—	59.7	64.7

Table 11. Comparison of offline RL methods on the **D4RL Adroit** benchmark. We report mean normalized scores for all baselines, with ±std for competitive baselines. The **best mean score** is bolded, and marked methods are statistically similar under a t-test. Our results use 6 seeds, each evaluated at the final step with 20 episodes. In addition, we include the average dataset performance, denoted as $\pi_{\mathcal{D}}$, to provide a reference for **data quality**.

		Model-free				Conservative model-based						Ours
Dataset	$\pi_{\mathcal{D}}$	BC	EDAC	IQL	ReBRAC	MOPO	MOBILE	ARMOR	ROMI	MoMo	VIPO	NEUBAY
pen-human	88.7	71.0±6.3	52.1±8.6	78.5±8.2	**103.2**±8.5	10.7	30.1±14.6	72.8±13.9	63.8±8.4	74.9	52.6±7.7	20.8±13.2
pen-cloned	68.7	51.9±15.2	68.2±7.3	83.4±8.2	**102.8**±7.8	54.6	69.0±9.3	51.4±15.5	57.7±3.9	74.1	71.1±9.5	91.3±21.2
door-human	7.7	2.3±4.0	10.7±6.8	3.3±1.8	-0.1±0.0	-0.2	-0.2±0.1	6.3±6.0	0.0±0.0	**11.3**	2.0±0.3	0.0±0.0
door-cloned	3.2	-0.1±0.0	9.6±8.3	3.1±1.8	0.1±0.1	15.3	**24.0**±22.8	-0.1±0.0	14.3±5.8	5.8	—	0.0±0.0
hammer-human	1.5	**3.0**±3.4	0.8±0.4	1.8±0.8	0.2±0.2	0.3	0.4±0.2	1.9±1.6	2.2±0.2	1.7	1.1±0.9	0.0±0.0
hammer-cloned	0.7	0.6±0.2	0.3±0.0	1.5±0.7	5.0±3.8	0.5	1.5±0.4	0.7±0.6	0.0±0.0	0.7	2.1±0.2	**14.4**±9.7
AVG	N/A	21.5	23.6	28.6	**35.2**	13.5	20.8	22.2	23.0	28.1	—	21.1

Table 12. Comparison of offline model-based RL methods on the **D4RL AntMaze** benchmark. We report mean normalized scores for all baselines, with ±std for competitive baselines. The **best mean score** is bolded, and marked methods are statistically similar under a t-test. Our results use 6 seeds, each evaluated at the final step with 100 episodes. In addition, we include the average dataset performance, denoted as $\pi_{\mathcal{D}}$, to provide a reference for **data quality**.

Dataset	$\pi_{\mathcal{D}}$	CBOP	MOBILE	RAMBO	COMBO	MOBILE†	ROMI	ADMPO	LEQ	NEUBAY
umaze	28.8	0.0±0.0	0.0±0.0	25.0	80.3	77.0	70.3±11.4	88.4±1.2	**94.4**±6.3	66.1±20.1
umaze-diverse	1.0	0.0±0.0	0.0±0.0	0.0	57.3	20.4	32.8±13.0	**81.7**±8.6	71.0±12.3	74.4±12.1
medium-play	19.6	0.0±0.0	0.0±0.0	16.4	0.0	64.6	51.3±19.7	23.9±6.3	**58.8**±33.0	12.8±19.2
medium-diverse	11.3	0.0±0.0	0.0±0.0	23.2	0.0	1.6	30.1±10.0	24.1±5.7	**46.2**±23.2	19.4±12.4
large-play	10.5	0.0±0.0	0.0±0.0	0.0	0.0	2.6	0.0±0.0	8.3±4.1	**58.6**±9.1	0.0±0.0
large-diverse	10.6	0.0±0.0	0.0±0.0	2.4	0.0	7.2	6.4±4.2	0.0±0.0	**60.2**±18.3	0.5±1.1
AVG	N/A	0.0	0.0	11.2	22.9	28.9	31.1	37.7	**64.9**	28.9

Source of benchmarked baseline results. For the D4RL locomotion benchmark in Tab. 1, we adopt the results of CQL, EDAC, and MOPO from the MOBILE paper (Sun et al., 2023, Table 1), where the MOPO results correspond to the tuned variant (denoted as MOPO* therein). Results for the remaining baselines are taken directly from their respective original publications. For MAPLE, we report the improved variant that employs an ensemble size of 142 (instead of the default 14) as described in (Chen et al., 2021c), to enable a fairer comparison with our method, which uses an ensemble size of 100. Finally, we note that prior works may differ in the total number of gradient steps used for training compared to ours (2M steps in this benchmark); for example, MOPO, MOBILE, ADMPO, and VIPO report results after 3M steps, while LEQ, SUMO, and ROMI use 1M steps. We retain these numbers as reported, since we believe this setting reflects the most faithful comparison with previously published results.

For the NeoRL locomotion benchmark in Tab. 10, we use EDAC results from the MOBILE paper (Sun et al., 2023), CQL and MOPO from the NeoRL paper (Qin et al., 2022), and COMBO from the MoDAP paper (Choi et al., 2024). All other baselines are taken from their original publications. For MoDAP on Hopper-v3-High, we report the improved result obtained with an ensemble size of 40, as presented by the authors to demonstrate the benefit of larger ensembles.

For the D4RL Adroit benchmark in Tab. 11, we report BC, IQL, and ReBRAC results from the CORL paper (Tarasov et al., 2023b), and MOPO from the MOBILE paper (Sun et al., 2023). Results for other baselines are taken from their original publications. Standard deviations for MOPO and MoMo were not available, so we omit them.

For the D4RL AntMaze benchmark in Tab. 12, we report CBOP, MOBILE, RAMBO, COMBO, and LEQ from the LEQ paper (Park & Lee, 2025), as well as an alternative tuning of MOBILE, denoted MOBILE†, and ADMPO from the ADMPO paper (Lin et al., 2025). We omit standard deviations for RAMBO, COMBO, and MOBILE† since they are not reported in the corresponding source papers.

Statistical tests for benchmarking results. To assess statistical significance, we conduct Welch's one-sided t-test ($p < 0.05$). In the reported tables, we highlight all methods whose average scores are not significantly different from the best-performing method.

G.2. Full Results on Value Overestimation and Horizon Scales

Fig. 6–Fig. 9 report the full ablation results on truncation thresholds to complement Fig. 1, with the maximum rollout horizon for each batch of training rollouts shown in the last columns.

A key observation is that using a quantile threshold of $\zeta = 0.9$ corresponds *roughly* to a horizon cap of 1–10, which mirrors the short fixed horizons commonly adopted in prior work. This shows that such prior choices are not suitable for guiding the design of Bayesian RL, where long horizons are essential.

Summary on the effect of ζ. We further count the number of complete failures (i.e., scores ≤ 5.0) under different thresholds using Tab. 13. With $\zeta = 0.9$, 16 datasets fail; with $\zeta = 0.99$, 6 datasets fail; and with $\zeta = 0.999$, none fail. This suggests that a safe range for ζ lies between 0.999 and 1.0. Although $\zeta = 0.999$ often performs similarly to 1.0 (as the resulting adaptive horizons are close), we observe clear advantages of 1.0 on tasks such as D4RL walker2d-random-v2 and pen-cloned-v1. **Thus, we recommend using $\zeta = 1.0$ as a starting point for our algorithm.**

Summary on horizon scales. Although NEUBAY uses adaptive horizons, we report the empirical horizon scales (75th-percentile and maximum) to illustrate the typical horizon length required under our Bayesian formulation. In D4RL and NeoRL locomotion tasks ($T = 1000$), NEUBAY uses 75th-percentile horizons of 2^4-2^6 in 4 tasks, 2^7 in 5 tasks, 2^8 in 9 tasks, and 2^9 in 3 tasks; the corresponding maximum horizons of 2^6-2^8 in 4 tasks, 2^9 in 4 tasks, 1000 in 11 tasks. In Adroit ($T = 100$ or 200), the 75th-percentile horizon is 2^6-2^7, and the maximum horizon reaches the episode length T. In AntMaze ($T = 700$ or 1000), the 75th-percentile horizon is 2^4-2^8; the maximum horizon reaches 2^8-2^9. Overall, these statistics show that NEUBAY selects 75th-percentile rollout horizons of **64-512 steps** and maximum horizons of **256-1000 steps**, in 21 out of 23 tasks with $T = 1000$.

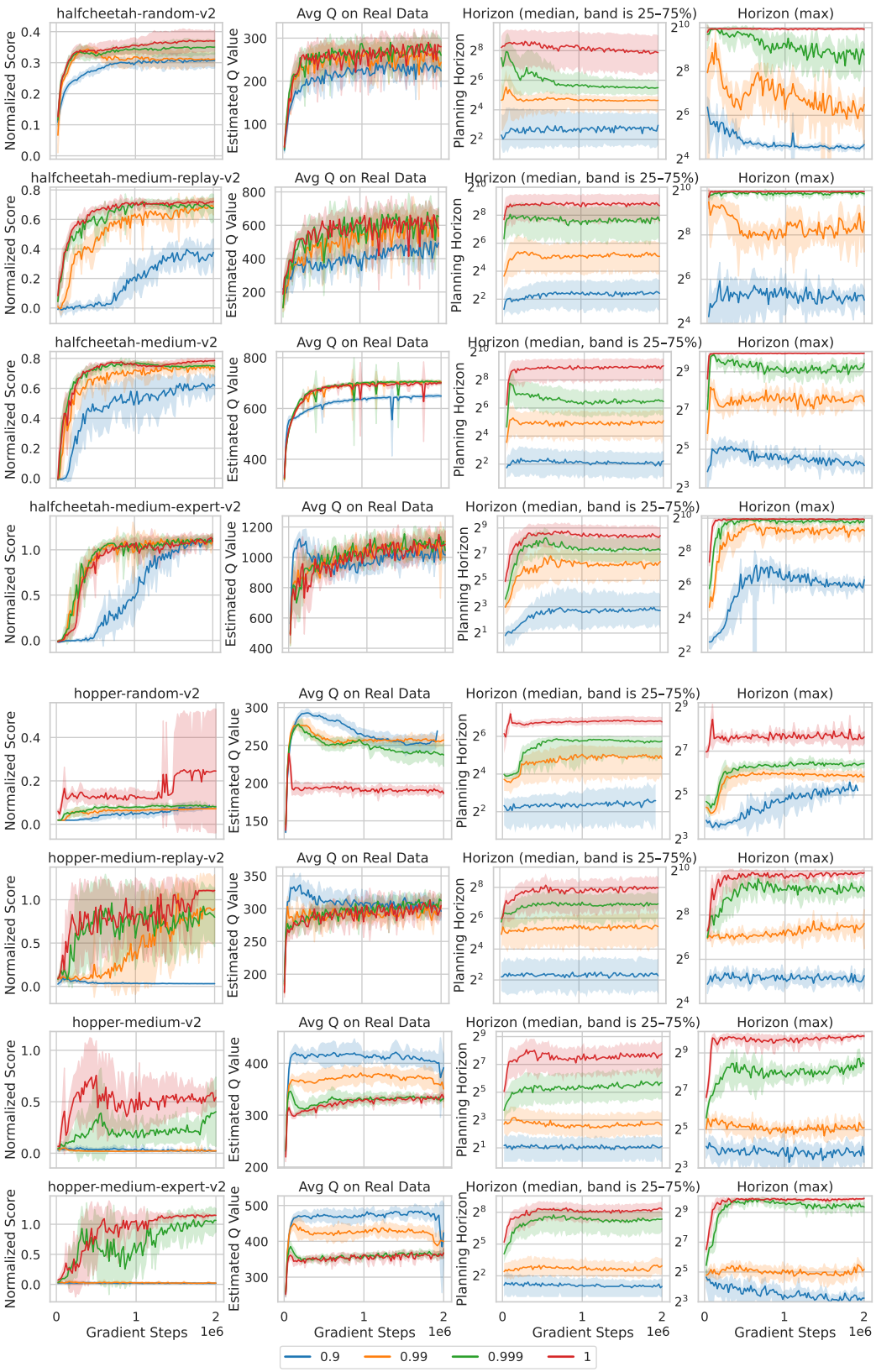

Figure 6. Ablation on the uncertainty quantile ζ for rollout truncation (part 1 of 4). The third column reports rollout horizon statistics (median with interquartile range) over 100 training-time rollouts, and the fourth column reports the maximum horizon.

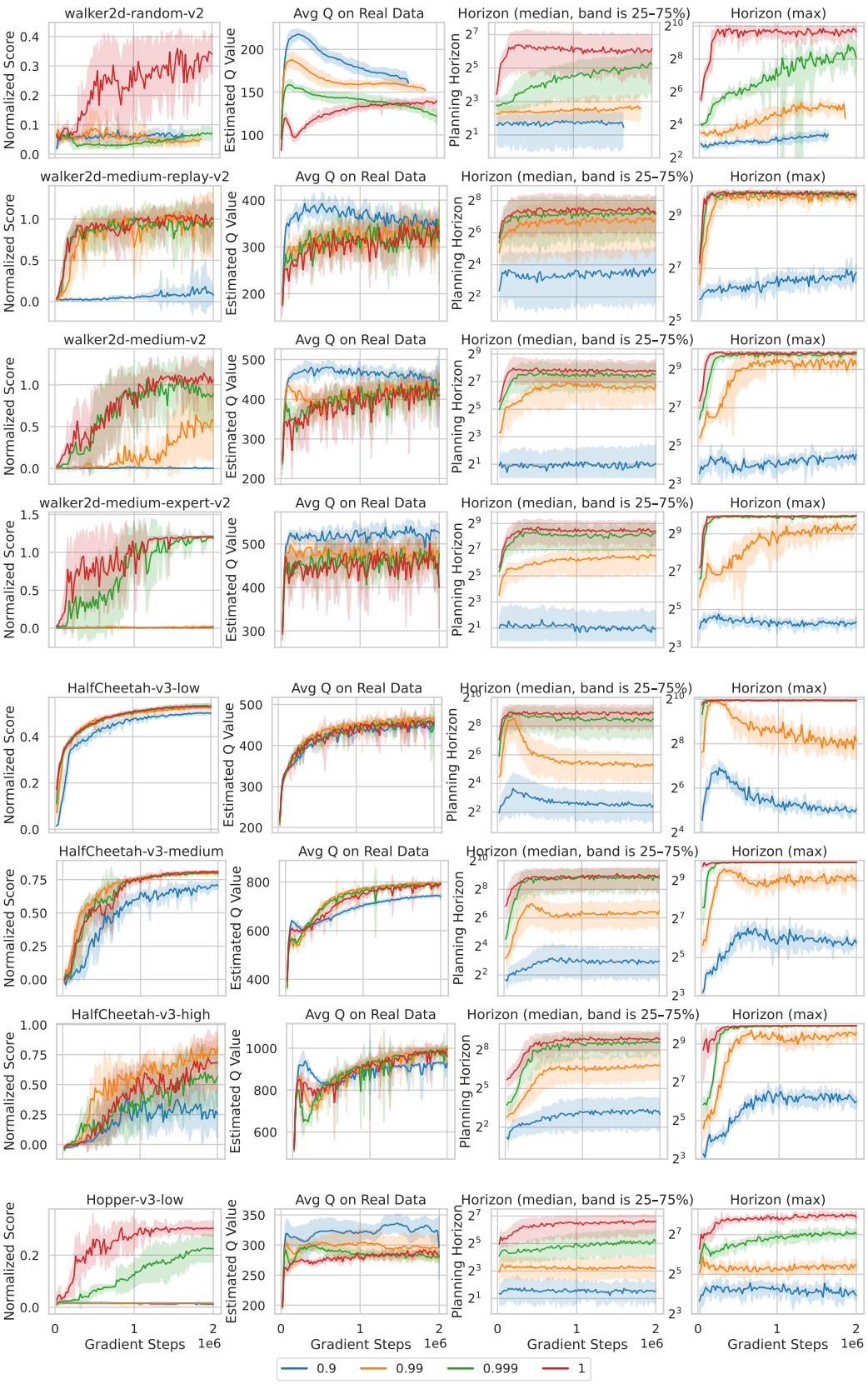

Figure 7. Ablation on the uncertainty quantile ζ for rollout truncation (part 2 of 4).

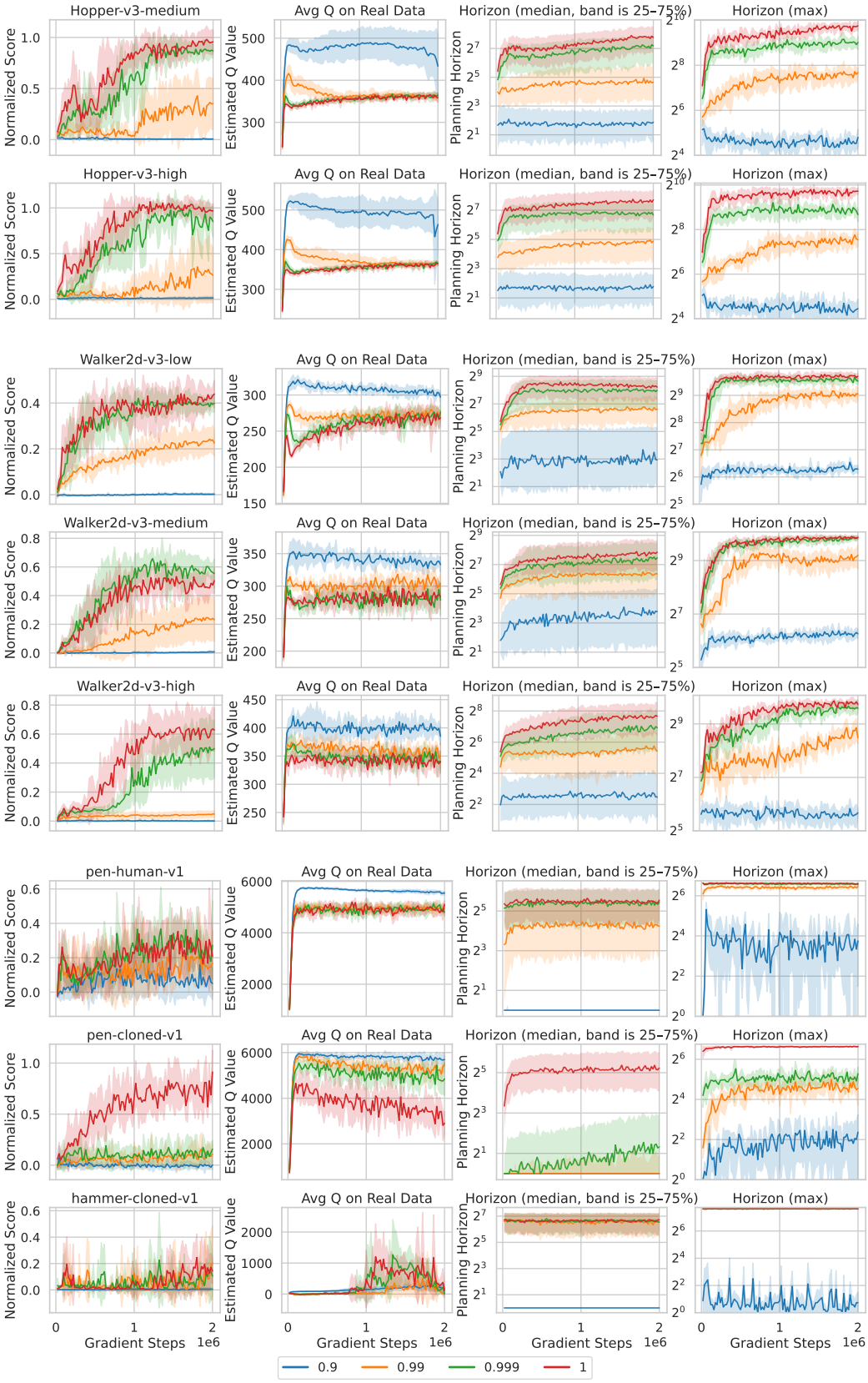

Figure 8. Ablation on the uncertainty quantile ζ for rollout truncation (part 3 of 4). Adroit benchmark has short maximum episode steps: $T = 100 < 2^7$ in pen and $T = 200 < 2^8$ in hammer, which limits the rollout horizon.

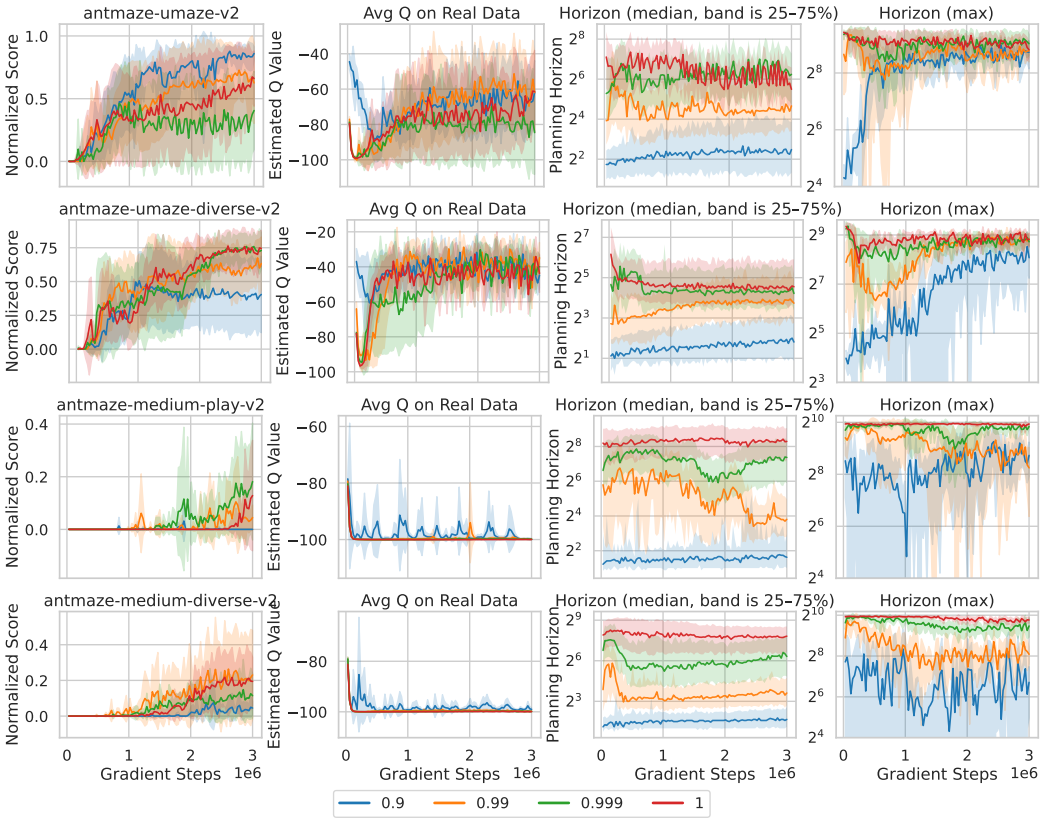

Figure 9. Ablation on the uncertainty quantile ζ for rollout truncation (part 4 of 4). Maximum episode steps are $T = 700$ in umaze and $T = 1000$ in medium maze. Successful episodes terminate early in AntMaze, so horizon lengths are partially confounded by this effect.

G.3. Full Results on Compounding Errors

Fig. 10–Fig. 12 report the full results on compounding errors to complement Fig. 5 in the main paper.

Plotting setup. For each world ensemble trained given a dataset, we evaluate compounding errors using *three* datasets that share the same underlying MDP ($\star$-random-v2, $\star$-medium-replay-v2, and $\star$-medium-expert-v2). For each evaluation dataset, we collect 200 rollouts in total (two rollouts per ensemble member). This evaluation protocol allows us to span a broad range of exploratory behaviors, similar to Zhou et al. (2025).

We generate a synthetic rollout as follows: first we draw $m_\theta \sim \mathbf{m}_{\boldsymbol{\theta}}$ and an entire real trajectory $(s_{0:T}, a_{0:T-1}, r_{1:T})$ from the evaluation dataset. We then let $\hat{s}_0 = s_0$, and $(\hat{r}_{t+1}, \hat{s}_{t+1}) \sim m_\theta(\hat{s}_t, a_t), \forall t < T$. Synthetic rollouts are truncated *only* when numerical overflow occurs (`float32`); we do not apply an uncertainty threshold and we ignore the terminal function. Because rollout lengths vary in Hopper, we apply forward filling (`pandas.DataFrame.ffill`) so that medians and percentile statistics remain well-defined. For the leftmost two columns, RMS denotes the root-mean-square, $\mathrm{RMS}(x) = \sqrt{\frac{1}{k}\sum_{i=1}^{k} x_i^2}$ for $x \in \mathbb{R}^k$, which normalizes the ℓ_2 norm to be dimension-invariant. RMSE denotes the root-mean-square error, $\mathrm{RMSE}(x, y) = \mathrm{RMS}(x - y)$ for $x, y \in \mathbb{R}^k$. For the rightmost scatter plot, we aggregate all state-action pairs from these rollouts and show the relation between estimated uncertainty $U_{\boldsymbol{\theta}}(\hat{s}_t, a_t)$ and the next-state error $\mathrm{RMSE}(\hat{s}_{t+1}, s_{t+1})$.

LayerNorm significantly suppresses worst-case (95%) errors across all evaluation setups. Across the $6 \times 3 = 18$ evaluation setups shown in Fig. 10–Fig. 12, LayerNorm consistently prevents compounding state-error and reward-bias explosion (1st and 3rd columns) by stabilizing the predicted state norm (2nd column). In contrast, without LayerNorm, **16 of the 18** setups exhibit clear error explosion. We further observe that models trained on medium and medium-expert datasets (with narrower coverage) tend to explode more rapidly than those trained on medium-replay datasets (with broader coverage). The only two non-exploding cases without LayerNorm occur when models trained on medium-replay are evaluated with random action sequences, likely because medium-replay offers the broadest coverage and includes random-action trajectories.

LayerNorm also significantly suppresses medium-case errors. LayerNorm effectively controls medium-case errors across all evaluation setups. In contrast, without LayerNorm, **12 of the 18** setups exhibit medium-case error explosion (with the remaining 6 showing comparable performance to the LayerNorm variant). Notably, although models trained on random datasets do not explode under random-action rollouts, they *do* explode under higher-quality action sequences when LayerNorm is removed.

Uncertainty threshold can safeguard with LayerNorm against compounding error. For world ensembles with LayerNorm, we find that the uncertainty threshold $\zeta = 1.0$ (used in our main experiments) reliably separates severe error regions, since $\zeta = 1.0$ approximates the boundary of in-distribution data. While the Spearman's rank coefficient, often used to benchmark uncertainty estimation accuracy (Lu et al., 2021), is less than 0.6 in 5 out of the 18 setups, we argue that this metric is less critical for Bayesian RL. Unlike approaches that penalize with uncertainties at every step, we only use uncertainty as a binary cutoff for truncation. As a result, our method is inherently more robust to imperfect uncertainty ranking, requiring only that severe errors lie beyond the $\zeta = 1.0$ boundary.

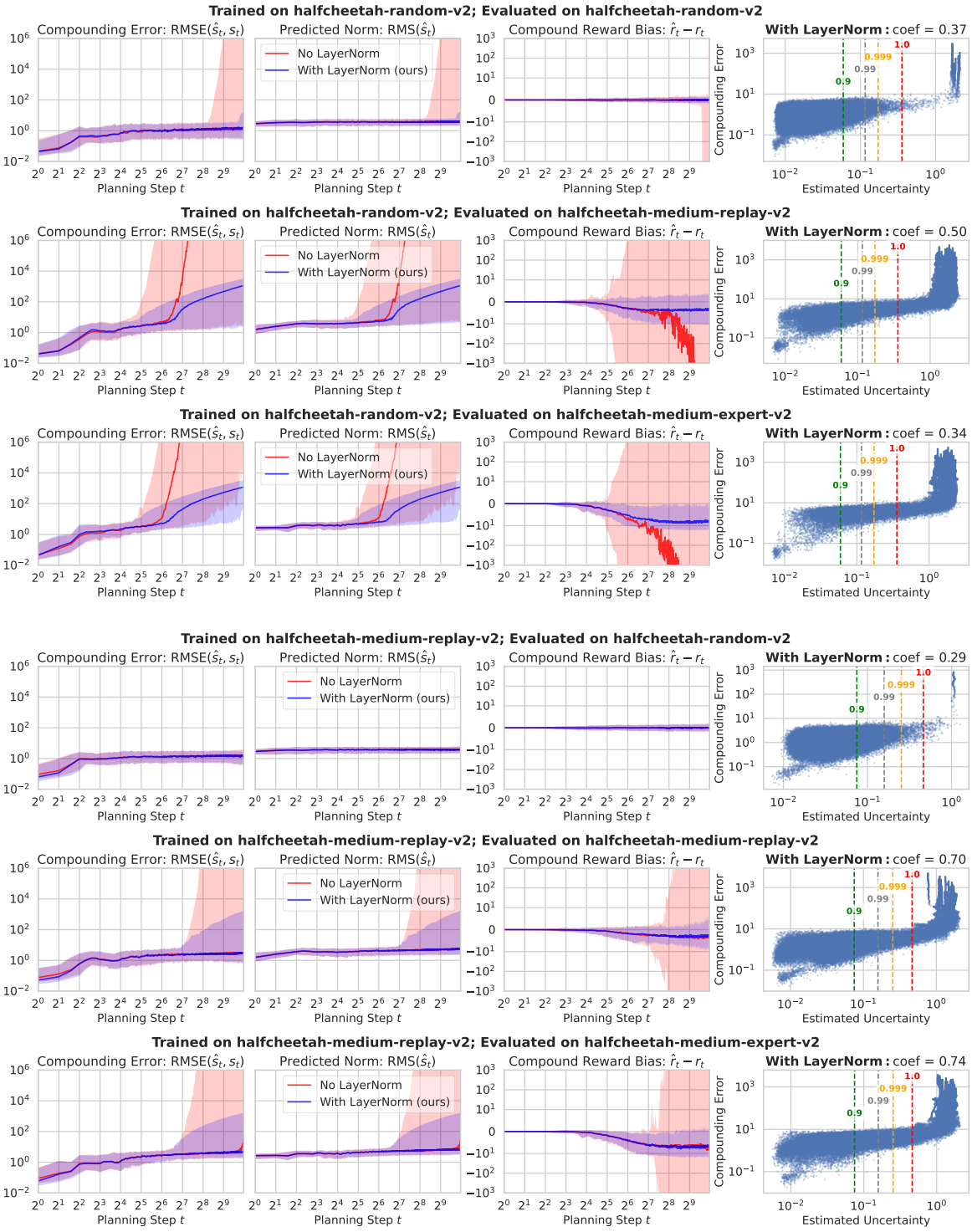

Figure 10. Effect of LayerNorm in world models (part 1 of 3). For each metric, we plot the **median** (solid line) together with the **5-95% percentile band** across 200 rollouts. The rightmost scatter plots show the Spearman's rank coefficients in the with-LayerNorm setting; vertical lines mark uncertainty thresholds $\zeta \in \{0.9, 0.99, 0.999, 1.0\}$.

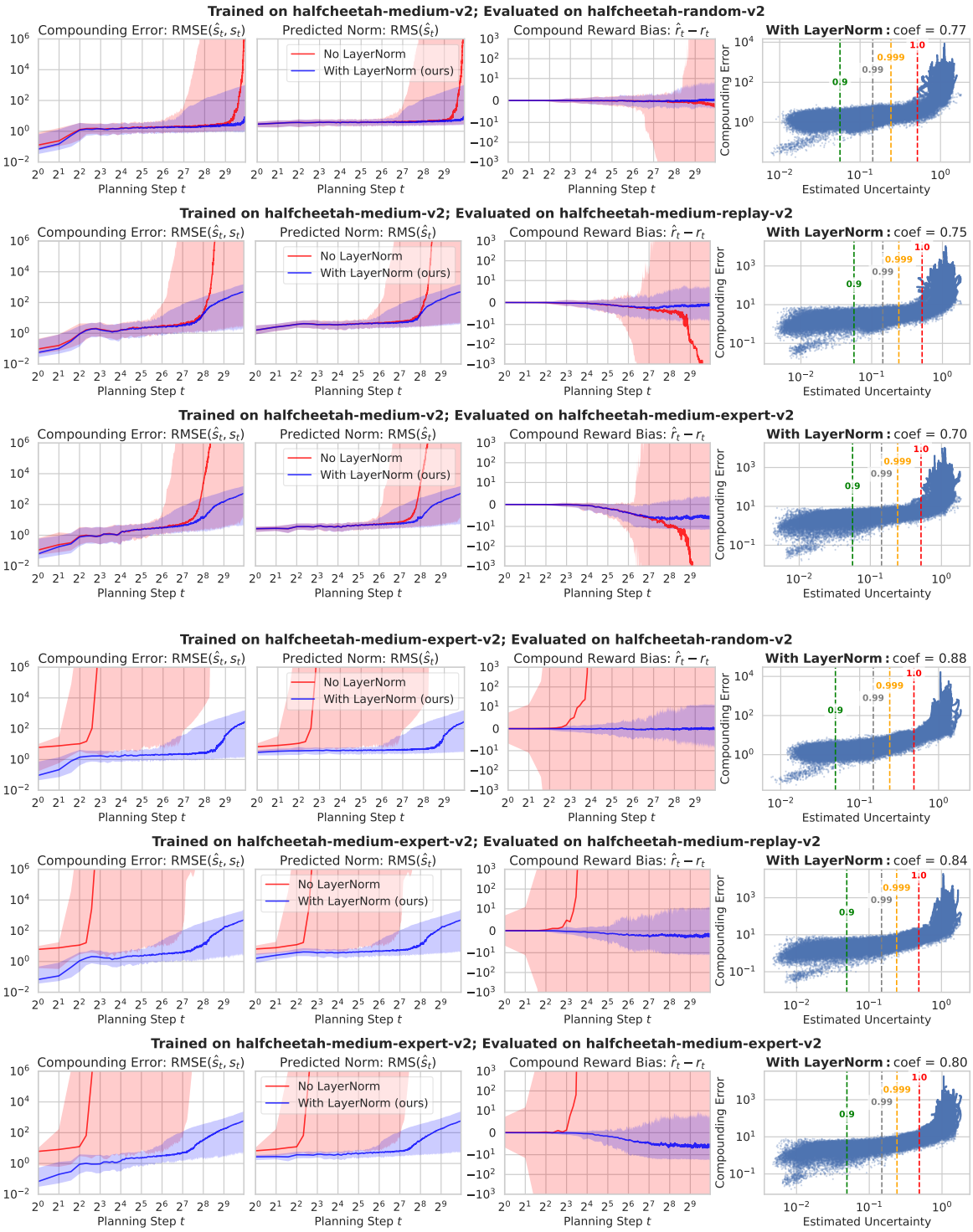

Figure 11. Effect of LayerNorm in world models (part 2 of 3).

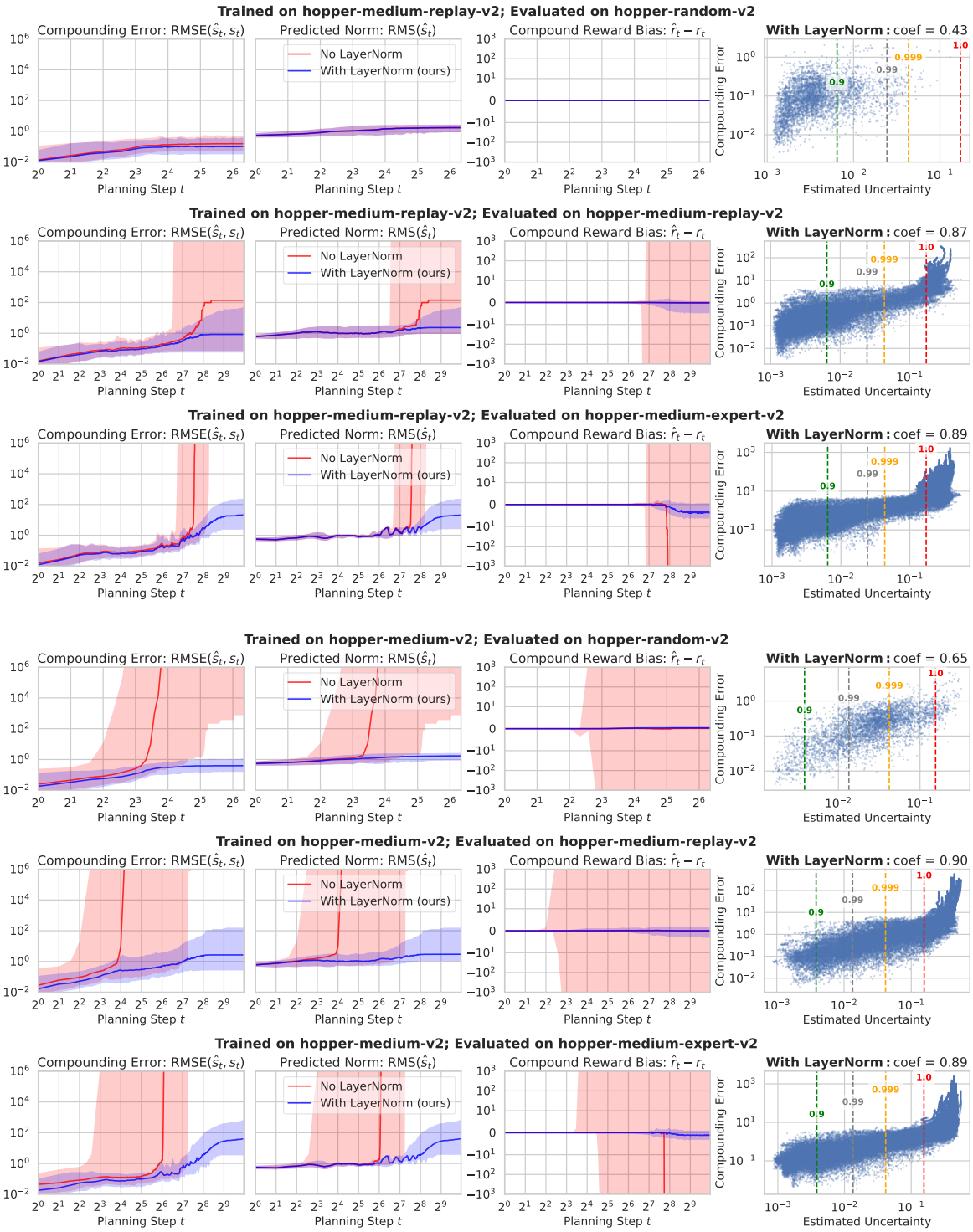

Figure 12. Effect of LayerNorm in world models (part 3 of 3).

G.4. Sensitivity and Ablation Results

Table 13. **Sensitivity and ablation results per dataset.** The highlighted setting (N=100, λ=0.0, ζ=1.0, using the entire history as agent input) is the main result. Ablations vary one hyperparameter at a time, except for the Markov agent, where we sweep the real data ratio κ for a fair comparison. Red shading shows **degradation** level: light (3–10), medium (10–30), dark (>30). Green shading shows **improvement** level: light (3–10), medium (10–30), dark (>30). Tab. 2 is a summary.

	Ensemble size N			Unc. penalty coef. λ					Truncation threshold ζ				Agent input	
Dataset	100	20	5	0.0	0.04	0.2	1.0	5.0	1.0	0.999	0.99	0.9	Hist.	Mark.
hc-random-v2	37.0	39.7	38.2	37.0	35.6	32.7	19.3	1.2	37.0	35.0	31.2	30.7	37.0	44.4
hp-random-v2	24.5	15.1	13.3	24.5	48.2	40.1	31.6	18.8	24.5	8.0	7.1	8.7	24.5	21.7
wk-random-v2	34.1	20.5	8.2	34.1	33.0	34.2	23.3	0.0	34.1	7.1	5.3	6.0	34.1	33.4
hc-med-rep-v2	72.1	68.6	66.7	72.1	72.0	70.1	63.5	52.1	72.1	67.8	68.6	37.7	72.1	76.9
hp-med-rep-v2	110.6	81.8	75.2	110.6	110.9	97.6	95.3	28.8	110.6	79.8	89.7	3.2	110.6	47.2
wk-med-rep-v2	99.3	91.3	97.8	99.3	87.4	93.5	86.7	72.8	99.3	96.0	92.3	7.4	99.3	112.8
hc-medium-v2	78.6	73.7	74.2	78.6	77.5	70.7	59.8	54.6	78.6	74.5	74.0	61.8	78.6	78.2
hp-medium-v2	54.2	37.1	48.8	54.2	52.2	105.8	74.0	64.8	54.2	40.3	2.3	2.2	54.2	47.0
wk-medium-v2	106.4	103.1	55.8	106.4	96.6	77.9	93.6	81.9	106.4	88.9	56.5	0.2	106.4	112.4
hc-med-exp-v2	109.4	107.3	97.8	109.4	107.7	112.7	110.3	96.4	109.4	112.3	109.6	109.6	109.4	111.9
hp-med-exp-v2	114.8	100.2	96.8	114.8	114.4	113.5	110.3	110.1	114.8	106.5	2.2	1.8	114.8	101.3
wk-med-exp-v2	120.6	120.6	118.5	120.6	121.5	116.2	109.3	107.4	120.6	118.5	2.0	1.1	120.6	122.2
hc-v3-Low	53.0	51.4	52.2	53.0	52.3	47.3	37.8	26.3	53.0	53.3	52.7	50.1	53.0	55.4
hp-v3-Low	30.3	26.6	29.3	30.3	30.0	24.3	15.5	12.7	30.3	22.5	1.6	1.1	30.3	29.4
wk-v3-Low	43.9	41.9	35.4	43.9	46.1	45.7	53.0	53.2	43.9	39.8	22.5	0.2	43.9	57.4
hc-v3-Med	81.1	80.5	77.3	81.1	81.2	79.3	68.1	56.6	81.1	79.5	79.9	70.4	81.1	82.5
hp-v3-Med	95.7	88.6	94.1	95.7	89.3	81.6	72.1	33.4	95.7	86.8	35.2	0.6	95.7	94.9
wk-v3-Med	50.5	34.4	39.9	50.5	43.9	50.2	44.8	40.9	50.5	55.5	23.1	0.7	50.5	47.1
hc-v3-High	68.3	59.9	53.8	68.3	65.9	67.5	71.1	54.1	68.3	56.7	80.4	25.2	68.3	81.5
hp-v3-High	96.8	100.4	99.3	96.8	89.6	75.3	90.2	46.4	96.8	85.0	26.1	1.6	96.8	80.9
wk-v3-High	62.7	59.1	57.0	62.7	55.3	67.2	72.2	58.5	62.7	50.0	4.8	0.0	62.7	72.4
pen-human-v1	20.8	19.3	13.7	20.8	25.8	24.0	35.9	34.8	20.8	27.2	16.8	5.0	20.8	28.0
pen-cloned-v1	91.3	63.4	75.6	91.3	68.2	76.0	77.2	67.2	91.3	15.7	10.6	0.2	91.3	75.6
hammer-cloned-v1	14.4	7.4	8.6	14.4	7.0	2.7	1.4	0.2	14.4	10.5	20.5	0.6	14.4	5.6
umaze-v2	66.1	71.1	65.3	66.1	19	9.3	16.7	28.4	66.1	40.6	62.6	86.0	66.1	0.0
umaze-diverse-v2	74.4	52.5	37.8	74.4	0.8	3.3	0.2	19.8	74.4	73.0	63.3	40.4	74.4	5.2
medium-play-v2	12.8	2.2	0.8	12.8	0.0	0.0	0.0	0.0	12.8	18.2	4.5	0.0	12.8	0.0
medium-diverse-v2	19.4	6.8	11.3	19.4	0.0	0.0	0.0	0.0	19.4	11.4	23.1	4.5	19.4	0.0

Ablation: Using a Markov agent in NEUBAY works well on most locomotion tasks, but fails in AntMaze. Popular offline RL methods employ Markov agents, while NEUBAY uses history-dependent agents due to its epistemic POMDP formulation. To assess the practical importance of memory, we replace the history-dependent agent in NEUBAY with a Markov one, sweep over the same real-data-ratio range, and report the best results in Tab. 2 and Tab. 13. The Markov version performs similarly to the history-dependent one on most *locomotion* tasks, especially on the NeoRL benchmark (64.7 → 66.8). However, it suffers severe degradation on hopper-medium-replay-v2 (110.6 → 47.2) and on all AntMaze tasks (43.2 → 1.3). This pattern suggests that in locomotion domains, epistemic uncertainty is relatively mild, so using a single observation may often infer the model index, making memory less critical. As a result, NEUBAY with a Markov agent remains a strong baseline in these settings. In contrast, AntMaze has high epistemic uncertainty, especially about the maze layout which is crucial to navigation, so memory is needed to accumulate information over time.

Table 14. Ablation results on using **fixed** horizon lengths $H \in \{256, 10, 3\}$ instead of adaptive long horizon (highlighted). For the Adroit domain which has maximum episode steps ≤ 200, we use $H = 32$ to replace $H = 256$. Red shading shows **degradation** level: light (3–10), medium (10–30), dark (>30). Green shading shows **improvement** level: light (3–10), medium (10–30), dark (>30).

Dataset	Adaptive H	$H = 256$	$H = 10$	$H = 3$
hc-random-v2	37.0	40.6	27.8	24.4
hp-random-v2	24.5	11.7	8.5	8.2
wk-random-v2	34.1	16.9	4.1	8.3
hc-med-rep-v2	72.1	73.1	54.9	4.0
hp-med-rep-v2	110.6	95.1	3.8	3.8
wk-med-rep-v2	99.3	92.6	15.7	7.4
hc-medium-v2	78.6	75.8	66.9	41.9
hp-medium-v2	54.2	39.4	4.8	2.2
wk-medium-v2	106.4	60.2	4.5	2.4
hc-med-exp-v2	109.5	108.0	98.7	71.2
hp-med-exp-v2	114.8	115.8	3.2	1.7
wk-med-exp-v2	120.7	118.4	4.5	2.2
hc-v3-Low	53.1	52.5	50.7	44.0
hp-v3-Low	30.3	39.4	2.6	1.5
wk-v3-Low	43.9	35.6	4.6	1.0
hc-v3-Med	81.1	79.8	71.3	51.1
hp-v3-Med	95.7	96.0	6.2	0.8
wk-v3-Med	50.5	49.7	6.0	0.0
hc-v3-High	68.3	68.6	59.5	28.7
hp-v3-High	96.8	98.4	3.2	1.1
wk-v3-High	62.7	45.3	4.5	2.2
pen-human-v1	20.8	3.5	8.6	0.2
pen-cloned-v1	91.3	86.2	52.1	6.7
hammer-cloned-v1	14.4	17.8	1.1	0.2
umaze-v2	66.1	69.9	71.8	33.4
umaze-diverse-v2	74.4	39.0	73.3	24.9
medium-play-v2	12.8	10.7	0.0	0.0
medium-diverse-v2	19.4	13.2	0.8	0.0

Ablation: Fixed short horizons collapse performance, and fixed long horizons underperform adaptive horizons. As shown in Tab. 14, fixed short horizons ($H = 3, 10$), which are common in offline MBRL, fail dramatically in our setting. Their performance drops across most datasets, consistent with our analysis that short rollouts increase reliance on bootstrapping and thus exacerbate value overestimation. Using a fixed long horizon ($H = 256$; or $H = 32$ in Adroit) alleviates this issue and is often competitive, but still falls short of adaptive truncation overall. These results confirm that fixed short horizons are harmful to non-conservative RL, and adaptive long horizons are better than fixed long horizons.

Table 15. Ablation results on using **SAC** (Haarnoja et al., 2018a) instead of REDQ (Chen et al., 2021b) as the backbone actor-critic. The highlighted setting (REDQ, ζ=1.0) is the main result. Red shading shows **degradation** level: light (3–10), medium (10–30), dark (>30). Green shading shows **improvement** level: light (3–10), medium (10–30), dark (>30).

Dataset	REDQ (ζ=1.0)	SAC (ζ=1.0)	SAC (ζ=0.9)
hc-random-v2	37.0	36.8	31.7
hp-random-v2	24.5	15.7	8.0
wk-random-v2	34.1	25.0	4.3
hc-med-rep-v2	72.1	70.4	34.2
hp-med-rep-v2	110.6	98.9	4.9
wk-med-rep-v2	99.3	61.5	20.6
hc-medium-v2	78.6	75.9	62.9
hp-medium-v2	54.2	42.4	2.1
wk-medium-v2	106.4	80.2	0.9
hc-med-exp-v2	109.5	108.8	111.0
hp-med-exp-v2	114.8	109.3	2.0
wk-med-exp-v2	120.7	118.9	0.7

Ablation: The dominant factor is horizon length, rather than REDQ vs. SAC. Tab. 15 shows that replacing REDQ with standard SAC only causes a moderate drop in performance when long horizons are retained (ζ=1.0), whereas enforcing short horizons with SAC (ζ=0.9) leads to a dramatic collapse on most datasets. This indicates that the main gain of NEUBAY does not come from the specific choice of REDQ, but from enabling sufficiently long rollouts. REDQ still provides a useful stabilization benefit, but it is secondary compared with the effect of horizon length.

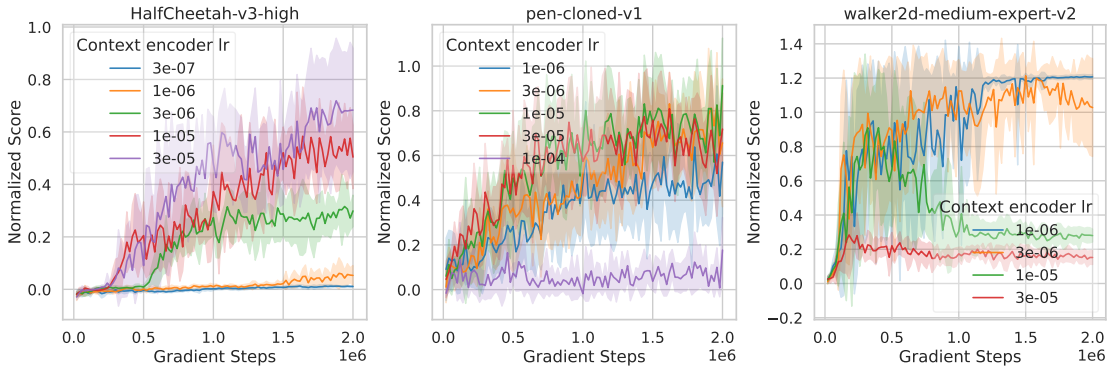

Figure 13. Selective learning curves on datasets where performance is *sensitive* to the **context encoder learning rate**, favoring high (left), medium (middle), and low (right) values.

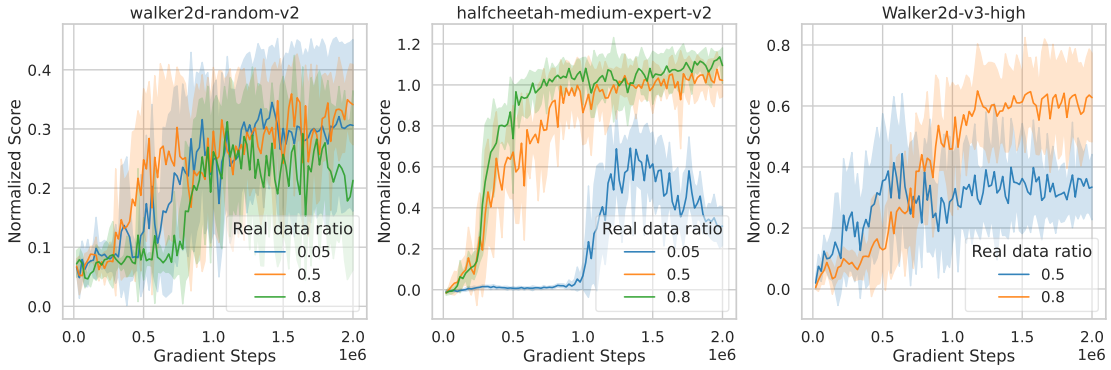

Figure 14. Selective learning curves on datasets where performance is *sensitive* to the **real data ratio**.

G.5. Failure-Case Analysis in AntMaze

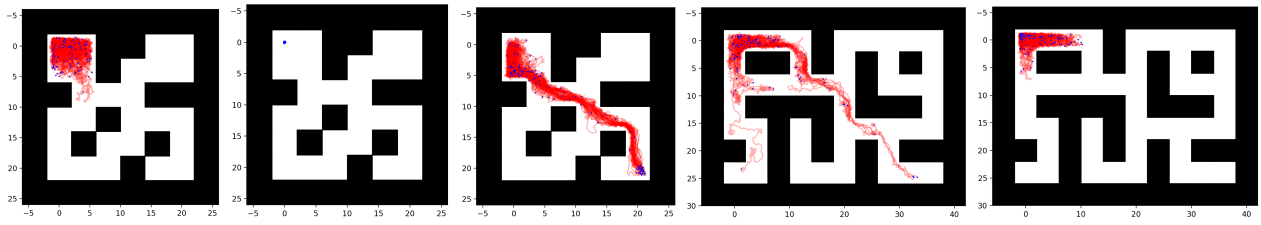

Figure 15. Failure cases in antmaze-medium-play-v2 (left three; different seeds) and antmaze-large-diverse-v2 (right two). The agent starts at the top-left corner and the goal is at the bottom-right. We show 100 evaluation trajectories in red and mark their endpoints in blue.

Fig. 15 illustrates typical failure cases of NEUBAY in AntMaze. Due to sparse rewards, the agent receives no learning signal before reaching the goal, often remaining near the initial region or colliding with nearby walls. Introducing explicit conservatism, such as uncertainty penalties, further degrades performance (Tab. 13). We attribute these failures primarily to limited exploration in large mazes and long-horizon modeling challenges in contact-rich dynamics. Addressing these issues likely requires stronger planning and world modeling components.