LEVERAGING GENERATIVE UNCERTAINTY MODELING FOR DISTRIBUTIONALLY ROBUST OFFLINE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Distributionally robust offline RL addresses the challenge of distribution shift of deployment environment by optimizing policies against an uncertainty set of environment dynamics. The uncertainty-set design is critical: it should capture diverse distribution shifts for robustness while excluding unrealistic shifts that lead to overly pessimistic policies. Existing methods often assume rectangular uncertainty, which can combine locally worst-case transitions that cannot coexist in a single environment. Many also rely on standard distribution discrepancy measures: ϕ-divergences ignore support-space geometry, while Wasserstein sets can be computationally challenging in high-dimensional continuous settings. To capture non-rectangular, diverse yet plausible shifts, we construct a generative uncertainty set using flow models which can generate diverse and plausible trajectories. The proposed set captures temporally coupled dynamics shifts through shared model parameters while maintaining consistency with offline data via a tractable flow-matching constraint that implies a Wasserstein bound. Building on this uncertainty model, we develop Generative Uncertainty-Augmented Robust Offline RL (`GURO-RL`), which addresses value evaluation and adversarial optimization under flow-based uncertainty and finite offline data. Experiments across multiple applications demonstrate that `GURO-RL` improves policy robustness under diverse distribution shifts.

1 INTRODUCTION

Offline reinforcement learning (RL) learns policies from a fixed dataset without further environment interaction (Levine et al., 2020; Prudencio et al., 2023; Kostrikov et al., 2021; Kumar et al., 2020; Fujimoto et al., 2019), making it attractive when online exploration is costly or impractical, such as in robotics, industrial control, and infrastructure systems. Its key challenge is that the offline dataset provides only finite coverage of the training environment, while deployment dynamics may shift, causing policy degradation under out-of-distribution (OOD) conditions.

Distributionally robust offline RL methods (Blanchet et al., 2023; Shi & Chi, 2024; Zhang et al., 2025; Liu & Xu, 2024; Wang et al., 2024; Panaganti et al., 2022) address the distribution shift issues by optimizing policies against an uncertainty set (a.k.a. ambiguity set) of transition-reward distributions. The uncertainty-set design is critical: it should be broad enough to capture meaningful distribution shifts, yet sufficiently constrained to exclude unrealistic environments. An overly narrow set provides limited robustness, whereas an overly broad set yields overly pessimistic policies. Its structure also directly determines the tractability of robust policy training.

Many robust offline RL methods Blanchet et al. (2023); Ma et al. (2022); Shi & Chi (2024); Zhou et al. (2021); Liu & Xu (2024) impose (s, a)- or s-rectangular uncertainty, enabling tractable worst-case value evaluation through robust Bellman operators Xu & Mannor (2010); Smirnova et al. (2019). However, practical distribution shifts are often coupled by shared latent factors, and rectangular sets may combine locally worst-case transitions that cannot coexist in one environment, leading to overly pessimistic policies. Recent studies Li et al. (2025); Wang & Si (2026); Li et al. (2026); Wang et al. (2025); Satheesh et al. (2026) advance non-rectangular robust RL, but general frameworks are still needed to leverage generative models for capturing diverse yet plausible shifts.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Method	Generative model	Representation	Discrepancy metric	Objective	Adversarial search
ROMB (Chen et al., 2026)	Explicit probability	Transition kernel	KL divergence	Worst-case value	Re-training
PLR-PVL (Berdica et al., 2024)	Fully-connected MLP	Next observation/reward	L_2 loss	Worst-case value	Finite search
AD-RRL (Foffano et al., 2025)	Implicit (Diffusion)	State-action-reward trajectory	N/A	CVaR	Guidance
GURO-RL (This work)	Implicit (Flow)	State-reward trajectory	Flow-matching loss	Worst-case value	Re-training

Table 1: Comparison of generative-model-based robust offline RL studies.

The effectiveness of an uncertainty set also depends on its distribution discrepancy metric. Uncertainty sets based on KL and other ϕ-divergences Liu et al. (2022); Chen et al. (2025); Hu & Hong (2013) enable tractable worst-case search but typically restrict distribution shifts to the support of the training distribution. Total-variation (TV) based uncertainty sets Wang et al. (2024); Lu et al. (2024) allow support mismatch but ignore support space geometry, potentially admitting highly unrealistic samples. Wasserstein sets Xu et al. (2023); Abdullah et al. (2019); Mohajerin Esfahani & Kuhn (2018) capture both probability-mass and support shifts, but can be computationally challenging in high-dimensional continuous spaces and often require conservative approximations.

To capture non-rectangular, diverse yet plausible distribution shifts, we exploit implicit generative models, such as diffusion and flow models Lipman et al. (2023); Luo (2022), to construct the uncertainty set. Due to their capability to generate realistic and high-dimensional samples, generative models have been increasingly used for RL world modeling Alonso et al. (2024); Ding et al. (2024); Jiang et al. (2026); Ye et al. (2026), but these approaches often do not explicitly account for deployment distribution shifts. Recent studies have begun incorporating generative models into robust RL. Chen et al. (2026) consider an uncertainty set defined by an explicit probabilistic model, whose method does not apply to implicit generative models. Berdica et al. (2024) model the transition-reward with conventional MLPs. Foffano et al. (2025) focus on optimizing Conditional Value-at-Risk (CVaR) by adversarially diffusion model guidance. Table 1 summarizes the differences.

By contrast, we leverage expressive and efficient flow models to construct a generative uncertainty set using flow-matching loss as the discrepancy measure. This formulation is well suited to robust RL because it captures distributions of diverse, realistic and high-dimensional samples while implying a Wasserstein distributional constraint. We further develop Generative Uncertainty-Augmented Robust Offline RL (GURO-RL), which addresses the challenges of value evaluation under flow-modeled uncertainty based on a finite offline dataset. Our contributions are summarized as follows:

- We build a generative uncertainty set for MDP reward-transition dynamics based on conditional guided flow models, which expresses non-rectangular reward-transition uncertainty and implies a Wasserstein uncertainty set by proof.
- We develop GURO-RL, a distributionally robust offline RL framework that adversarially fine-tunes the flow world model within the generative uncertainty set and incorporates the generated adverse trajectories into conservative value-based offline RL, thus addressing both distribution shifts and limited offline coverage.
- We evaluate GURO-RL under multiple applications with diverse OOD conditions on MuJoCo environments and an EPANET water-distribution-network case study, demonstrating improved robustness across nominal and shifted environments.

2 Related Work

Robust Reinforcement Learning. Robust RL optimizes policies against worst-case environment models (Iyengar, 2005; Wiesemann et al., 2013; Gu et al., 2025; Nilim & El Ghaoui, 2003), often using state- or state–action-rectangular uncertainty for tractable Bellman updates. However, rectangular sets may combine incompatible local perturbations and become overly conservative. In robust offline RL, methods are developed to update robust value under total-variation, norm-bounded, or other transition uncertainty models (Panaganti et al., 2022; Zhang et al., 2025; Wang et al., 2024; Levine et al., 2020; Ma et al., 2022). Recent work further studies non-rectangular uncertainty through data-driven ambiguity sets, robust policy gradients, and structured transition

models (Li et al., 2025; Wang et al., 2025; Satheesh et al., 2026). By comparison, we construct a flow-based non-rectangular ambiguity set over joint multi-step state–reward distributions. Shared flow parameters couple distribution shifts across state–action conditions, while a flow-matching constraint anchors adversarial adaptation to the offline data.

Generative Models For RL. Generative models play two main roles in RL: policy/planner representation and environment modeling. For the former, diffusion planners generate trajectories conditioned on returns or constraints (Janner et al., 2022; Ajay et al., 2023), while flow-based policies support guided generation and policy optimization (Lipman et al., 2023; Zheng et al., 2023; McAllister et al., 2025; Ye et al., 2026). For environment modeling, generative models predict multi-step states and/or rewards for offline value estimation and policy improvement (Alonso et al., 2024; Ding et al., 2024; Jiang et al., 2026; Ye et al., 2026). Recent studies further incorporate generative environment modeling into robust RL as in Table 1. Chen et al. (2026) consider an uncertainty set defined by an explicit probabilistic model and derive KL-constrained Stackelberg optimization, which does not apply to the considered implicit generative models. Berdica et al. (2024) model transition-reward dynamics with conventional MLPs and search for worst-case kernels over a finite set of well-trained models based on squared error, while Foffano et al. (2025) optimize CVaR using adversarially guided diffusion. In contrast, we construct a generative uncertainty set over multi-step state–reward distributions using flow models, enabling the representation of diverse yet plausible environment shifts.

3 Preliminary

3.1 Offline Reinforcement Learning

Consider a Markov Decision Process (MDP) (Puterman, 1994; Sutton & Barto, 2018) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \nu, d_0, \gamma)$ with the state space $\mathcal{S}$ and the action space $\mathcal{A}$, $\nu : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{R} \times \mathcal{S})$ denotes the MDP kernel of transition and reward with $\nu_{s,a}$ or $\nu(\cdot \mid s, a)$ denoteing the conditional transition probability, $d_0 \in \Delta(\mathcal{S})$ is the initial state distribution, and $\gamma \in (0, 1)$ is the discount factor.

Given a policy $\pi(a \mid s)$, the value function under an MDP with transition-reward kernel ν is defined as $V_\nu^\pi(s) = \mathbb{E}_{\pi,\nu}\left[\sum_{t=0}^{\infty} \gamma^t r_t \,|\, s_0 = s\right]$, where $a_t \sim \pi(\cdot \mid s_t)$ and $(r_t, s_{t+1}) \sim \nu(\cdot \mid s_t, a_t)$. The corresponding Q-function, which maps a state-action pair to its expected long-term return, is $Q_\nu^\pi(s, a) = \mathbb{E}_{(r,s')\sim\nu(\cdot|s,a)}\left[r + \gamma V_\nu^\pi(s')\right]$. Accordingly, the value function can also be written as $V_\nu^\pi(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}\left[Q_\nu^\pi(s, a)\right]$. Offline RL learns a policy from a fixed dataset of trajectories $\mathcal{D} = \left\{(s_{0,i}, a_{0,i}, r_{0,i}, \cdots, s_{|\tau_i|,i}, a_{|\tau_i|,i}, r_{|\tau_i|})\right\}_{i=1}^N$, collected by a behavior policy π_b, without further interaction with the environment (Levine et al., 2020). Its objective is $\max_\pi \mathbb{E}_{s_0\sim d_0}\left[V_\nu^\pi(s_0)\right]$.

3.2 Distributionally Robust Offline Reinforcement Learning

Deployment shifts in transitions or rewards can cause severe performance degradation for offline RL (Fujimoto et al., 2019; Kumar et al., 2020). Such shifts arise naturally in practice, e.g., from mismatched robot dynamics Muratore et al. (2022) or changes in demand, grid conditions, and device parameters in power systems Chen et al. (2022). To improve robustness under plausible deployment shifts, distributionally robust offline RL (Robust Offline RL) optimizes the policy for worst-case performance over an uncertainty set of transition-reward distributions consistent with the offline data (Panaganti et al., 2022; Xu & Mannor, 2010; Shi & Chi, 2024):

$$\max_\pi \min_{\nu\in\mathcal{U}} \mathbb{E}_{s_0\sim d_0}\left[V_\nu^\pi(s_0)\right], \tag{1}$$

where $\mathcal{U}$ is an uncertainty set of the transition-reward kernels.

Rectangularity Many existing studies adopt different rectangularity structures for the uncertainty set $\mathcal{U}$. Many consider (s, a)-rectangular sets, $\mathcal{U} = \prod_{(s,a)\in\mathcal{S}\times\mathcal{A}} \mathcal{U}_{s,a}$, where $\mathcal{U}_{s,a} \subseteq \Delta(\mathcal{S} \times \mathcal{R})$, which assume that transition-reward distributions are independent across state-action pairs (Shi & Chi, 2024; Clavier et al., 2024; Li & Shapiro, 2025). This yields the robust Bellman operator $(\mathcal{T}_{\text{rect}}^\pi V)(s) = \mathbb{E}_{a\sim\pi(\cdot|s)}\left[\inf_{\nu_{s,a}\in\mathcal{U}_{s,a}} \mathbb{E}_{(r,s')\sim\nu_{s,a}}[r + \gamma V(s')]\right]$, allowing worst-case evaluation to decompose across state-action pairs by locally searching adversarial distributions independently.

More generally, s-rectangular sets (Xu & Mannor, 2010; Wiesemann et al., 2013) allow coupling across actions at the same state while remaining independent across states: $\mathcal{U} = \prod_{s\in\mathcal{S}} \mathcal{U}_s$, where $\mathcal{U}_s \subseteq \{\mathcal{A} \to \Delta(\mathcal{S}\times\mathcal{R})\}$.

The independence assumptions of (s,a)- and s-rectangularity can be inappropriate in practice. For example, shared physical parameters such as robot mass, contact friction, or controller gain can jointly affect transitions across many states and actions in robot control (Peng et al., 2018). Rectangular uncertainty sets may therefore combine transition kernels that cannot coexist in a single environment, leading to overly conservative policies. We instead consider non-rectangular uncertainty sets, $\mathcal{U} \subseteq \{\mathcal{S}\times\mathcal{A} \to \Delta(\mathcal{S}\times\mathcal{R})\}$, without (s,a)- or s-rectangular factorization constraints.

Distributional discrepancy measure An uncertainty set is commonly defined around a nominal distribution $P_\circ$ as $\mathcal{U} = \{P : D(P, P_\circ) \le \epsilon\}$, where the distribution discrepancy measure D determines which distribution shifts are considered plausible. Many robust RL methods use ϕ-divergences, such as KL divergence (Liu et al., 2022; Shi & Chi, 2024; Chen et al., 2025), $D_\phi(P,Q) = \int \phi\Big(\frac{\mathrm{d}P}{\mathrm{d}Q}\Big)\,\mathrm{d}Q$, which often yield computationally convenient formulations but ignore the support space. They either restrict candidate distributions to the support of the training distribution, as with KL divergence, or ignore support-space geometry and may admit unrealistic shifts, as with TV distance. Wasserstein metrics (Dam et al., 2025; Xu et al., 2026; Coache & Jaimungal, 2026), $D_{\mathrm{W}}(P,Q) = \inf_{\gamma\in\Gamma(P,Q)} \big(\mathbb{E}_{(x,y)\sim\gamma}\|x-y\|^p\big)^{1/p}$, capture both probability-mass and support in distribution shifts, but they bring computational tractability issues for continuous support space and often require conservative approximations. In this work, we move beyond these conventional discrepancy measures and introduce a new discrepancy measure based on generative models.

3.3 Flow Model and Guided Flow Matching

Flow matching. A flow model transports a simple source distribution p_0, such as a standard Gaussian, to a data distribution p_1 through the ODE $\frac{\mathrm{d}X_k}{\mathrm{d}k} = u_k(X_k), k \in [0,1]$, where the vector field u_k induces a flow $X_k = \Phi_k(X_0)$ for $X_0 \sim p_0$, with intermediate distributions $p_k = (\Phi_k)_{\#}p_0$ (Lipman et al., 2023; 2024). Flow matching approximates u_k with a parameterized field $u_\theta(x,k)$ using tractable conditional probability paths indexed by endpoint pairs $Z = (X_0, X_1)$. Each path $p_k(x \mid Z)$ is associated with a conditional vector field $u_k(x \mid Z)$, whose marginal satisfies $u_k(x) = \mathbb{E}[u_k(X_k \mid Z) \mid X_k = x]$. For the linear interpolation $X_k = (1-k)X_0 + kX_1$, the target velocity is $X_1 - X_0$, yielding the conditional flow-matching loss $J_{\mathrm{CFM}}(\theta) = \mathbb{E}_{k\sim\mathrm{Unif}[0,1],\,(X_0,X_1)\sim q}\left[\|u_\theta(X_k,k) - (X_1-X_0)\|_2^2\right]$. The conditional and marginal flow-matching objectives differ only by a parameter-independent constant and therefore share the same population minimizers.

Guided flow models. In RL, generative models are needed to learn the conditional dynamic distribution $P(\cdot \mid s_t, a_t)$. Let $y = (s_t, a_t)$ denote the state-action condition. We model this distribution using guided flow matching (Zheng et al., 2023), which combines conditional flow matching with classifier-free guidance. To learn both conditional and unconditional vector fields with a single model, we sample $\zeta \sim \mathrm{Bernoulli}(p_{\mathrm{drop}})$ with $p_{\mathrm{drop}} \in (0,1)$ and define $\widetilde{Y} = \varnothing$ if $\zeta = 1$ and $\widetilde{Y} = y$ otherwise. The parameterized vector field $u_\theta(x,k,y)$, with $y \in (\mathcal{S}\times\mathcal{A}) \cup \{\varnothing\}$, is trained using

$$J_{\mathrm{GFM}}(\theta) = \mathbb{E}\left[\left\|u_\theta(X_k,k,\widetilde{Y}) - (X_1 - X_0)\right\|_2^2\right], \tag{2}$$

where the expectation is over the data (X_1, Y), source X_0, interpolation time k, and dropout variable ζ. During sampling, the flow model generates samples through the guided ODE $\frac{\mathrm{d}X_k}{\mathrm{d}k} = u_\theta^{\mathrm{gfm}}(X_k,k,y), X_0 \sim p_0$, where the guided vector field is

$$u_\theta^{\mathrm{gfm}}(x,k,y) = \lambda_{\mathrm{gfm}} u_\theta(x,k,y) + (1-\lambda_{\mathrm{gfm}}) u_\theta(x,k,\varnothing), \lambda_{\mathrm{gfm}} \ge 0. \tag{3}$$

Here, $\lambda_{\mathrm{gfm}} = 0$ gives the unconditional field, and increasing λ_{gfm} strengthens conditioning by extrapolating away from the unconditional field. In the following, we denote by $X_\theta : \mathcal{S}\times\mathcal{A} \to \mathcal{X}$ the sampling model and by $P_\theta : \mathcal{S}\times\mathcal{A} \to \Delta(\mathcal{X})$ the corresponding endpoint distribution of X_1.

4 GENERATIVE UNCERTAINTY MODELING BY FLOW MODELS

In Robust Offline RL (1), an ideal uncertainty set $\mathcal{U}$ should flexibly accommodate temporal dependence across transition-reward kernels, capture diverse yet plausible transition-reward distributions, and lead to tractable robust RL solutions. To achieve these goals, we construct the uncertainty set using a flow model.

Non-rectangular uncertainty set based on flow models. To capture dependencies among transition-reward distributions across state-action pairs, we introduce a conditional flow model $X_\theta : \mathcal{S} \times \mathcal{A} \to (\mathcal{S} \times \mathcal{R})^H$, which generates an H-step reward-state trajectory $\tau = (r_0, s_1, r_1, s_2, \ldots, r_{H-1}, s_H)$. Let $P_\theta(\tau \mid s, a)$ denote the implicit trajectory distribution induced by X_θ. When trained on the offline dataset $\mathcal{D}$ using guided flow matching, $P_\theta(\tau \mid s, a)$ approximates the offline trajectory distribution $P_\circ(\tau \mid s, a)$ induced by the transition-reward kernel $\nu(r, s' \mid s, a)$ and behavior policy π_b. In particular, when $H = 1$, $P_\theta(r_0, s_1 \mid s, a)$ reduces to a one-step transition-reward kernel ν. We define the generative uncertainty set in the flow-model parameter space as

$$\mathcal{U}_{\mathrm{G}} = \{P_\theta : D(P_\theta, P_\circ) \le \epsilon\}, \tag{4}$$

where D is a distributional discrepancy measure and ϵ controls the uncertainty-set size. Varying the shared parameter θ produces a family of trajectory distributions that remain close to the offline distribution. This shared parameterization naturally induces a non-rectangular uncertainty structure.

Distribution discrepancy measure based on flow matching. It remains to specify the discrepancy measure D in (4). Using a ϕ-divergence ignores the geometry of sample space and is limited in capturing meaningful support shifts. Moreover, because P_θ is defined implicitly by a flow sampler, such divergences can not be directly expressed by θ. Wasserstein metrics require solving the computationally difficult optimal transport problems on the parameterized space of the flow model.

Instead of relying on conventional discrepancy measures, we define the uncertainty set using the guided flow-matching loss as

$$\mathcal{U}_{\mathrm{G}} = \{P_\theta : J_{\mathrm{GFM}}(\theta) \le \epsilon\}, \tag{5}$$

where $J_{\mathrm{GFM}}(\theta)$ is given in (2). The guided flow-matching loss is ODE-simulation-free and can be efficiently computed, enabling efficient optimization without additional density estimation or reformulation. Moreover, Proposition 4.1 shows that $\mathcal{U}_{\mathrm{G}}$ is contained in the Wasserstein-2 uncertainty set $\mathcal{U}_W(P_\circ; \epsilon_W) = \left\{P_\theta : \mathbb{E}_Y\left[W_2^2(P_\theta(\cdot \mid Y), P_\circ(\cdot \mid Y))\right] \le \epsilon_W\right\}$, linking the tractable flow-matching constraint to an expressive measure of distributional shift.

Proposition 4.1. *Let Y be the state-action pair under the offline distribution and let $P_\theta(\cdot \mid Y)$ denote the flow-model endpoint distribution. Under Assumptions* (A1)–(A4) *in Appendix A, it holds that*

$$\mathbb{E}_Y\left[W_2^2(P_\theta(\cdot \mid Y), P_\circ(\cdot \mid Y))\right] \le e^{2L_{\mathrm{gfm}}} C_{\mathrm{gfm}} J_{\mathrm{GFM}}(\theta), \tag{6}$$

where W_2 is the Wasserstein-2 metric, L_{gfm} is the Lipschitz constant of the vector field model u_θ with respect to X, and $C_{\mathrm{gfm}} = \lambda_{\mathrm{gfm}}^2/(1 - p_{\mathrm{drop}}) + (1 - \lambda_{\mathrm{gfm}})^2/p_{\mathrm{drop}}$.

The proof of Proposition 4.1 is provided in Appendix A. The Proposition connects the flow-based uncertainty set in (5) to a Wasserstein uncertainty set (Abdullah et al., 2019) by showing that any $P_\theta \in \mathcal{U}_{\mathrm{G}}$ satisfies an averaged Wasserstein-2 distribution discrepancy bound with $\epsilon_{\mathrm{W}} = e^{2L_{\mathrm{cfg}}} C_{\mathrm{cfg}} J_{\mathrm{GFM}}(\theta)$. Thus, under the conditions of the Proposition, Robust Offline RL with the generative uncertainty set (5) remains distributionally consistent with the training distribution $P_\circ$. Meanwhile, the expressiveness of flow models enables $\mathcal{U}_{\mathrm{G}}$ to represent diverse yet plausible state-reward trajectories with high fidelity, capturing realistic deployment shifts. In the following sections, we show that this uncertainty set also leads to tractable Robust Offline RL algorithms.

5 GURO−RL ALGORITHM

In this section, we develop GURO−RL by incorporating the advantageous generative uncertainty set in (5) into Robust Offline RL.

Algorithm 1 GURO−RL: Generative Uncertainty-Augmented Robust Offline RL

Require: Offline dataset $\mathcal{D}$

1: Pretrain a policy $\pi = \pi_0$ with its value functions V_ψ^π, Q_ϕ^π and a flow model X_θ on $\mathcal{D}$
2: **for** each training step **do**
3: Finetune flow model X_θ by (11) to identify an adversarial model in the generative uncertainty set given policy π.
4: Sample a set $\mathcal{B}$ of $H-$step state-reward trajectories by X_θ based on $(s, a) \in \mathcal{D}$.
5: Update value functions Q_ϕ^π, V_ψ^π and the policy π on $\mathcal{B}$ by offline Q learning.
6: **end for**

5.1 Challenges of Generative Uncertainty Incorporation

Developing a Robust Offline RL algorithm typically requires addressing two key challenges. First, limited coverage of offline data causes extrapolation error in value estimation, as value estimates become unreliable for unseen state-action pairs. Second, distribution shifts in transition or reward distributions in deployment environment cause generalization error.

Many existing studies Blanchet et al. (2023); Shi & Chi (2024); Wang et al. (2024) address these challenges under rectangularity, enabling robust Bellman updates $Q(s,a) = \inf_{\nu_{s,a} \in \mathcal{U}_{s,a}} \mathbb{E}_{(r,s') \sim \nu_{s,a}} [r + \gamma V(s')]$. Some methods further assume linear MDPs Wang et al. (2024); Ma et al. (2022) or tabular state-action spaces Zhou et al. (2021), enabling closed-form confidence bounds for conservative value estimation. In contrast, our flow-based uncertainty set in (5) requires neither rectangularity nor linear or tabular assumptions. Consequently, robust Bellman decomposition and closed-form confidence bounds are unavailable, calling for new methods for worst-case environment search and conservative value estimation from finite offline data.

5.2 GURO−RL Framework

To address the challenges introduced by generative uncertainty, we develop GURO−RL in Algorithm 1, which separately handles distribution shift and finite offline coverage. GURO−RL first trains an initial policy π on $\mathcal{D}$ and estimates its value functions V_ψ^π and Q_ϕ^π in Line 1. Meanwhile, it trains a flow model X_θ on $\mathcal{D}$, thereby preserving the transition-reward information from the offline data.

We solve the distributionally robust problem through nested max–min optimization, alternately updating the flow model X_θ and policy π with competing value objectives. At each iteration in Line 3, given the current policy π, we adversarially fine-tune X_θ within the generative uncertainty set (5) to identify a trajectory distribution $P_\theta(\tau \mid s, a)$ that minimizes the H-step bootstrapped value:

$$\min_{\theta \in \mathcal{U}_G} \hat{V}_\theta^\pi(s),\ \hat{V}_\theta^\pi(s) = \mathbb{E}_{a \sim \pi(\cdot|s),\ \tau \sim P_\theta(\cdot|s,a)} \left[\sum_{h=0}^{H-1} \gamma^h r_h + \gamma^H V_\psi^\pi(s_H) \right], \tag{7}$$

where V_ψ^π is the value fitted in the previous iteration. When $H = 1$, the value expansion reduces to the standard Bellman update. Using $H > 1$ can benefit offline RL as it reduces reliance on the bootstrapped value $V_\psi^\pi(s_H)$ evaluated for state s_H possibly not covered by offline data. However, an overly large H can amplify distribution shift due to policy mismatch over long rollouts. Line 4 then samples adversarial trajectories using the resulting model P_θ based on $(s, a) \in \mathcal{D}$, which are stored in $\mathcal{B}$ to train the policy π with optimized value functions Q_ϕ^π and V_ψ^π in Line 5.

In the max–min iteration, both optimization steps suffer from limited offline coverage. Moreover, since the flow model is implicit, directly optimizing expectations under its induced distribution P_θ is challenging. We therefore reformulate (7) into tractable objectives that can be conservatively evaluated using the offline dataset, with a concrete optimization objective derived below.

5.2.1 Adversarial Flow-Model Finetuning

Value evaluation based on an offline dataset We first reformulate (7) for adversarial fine-tuning of the flow model. Direct evaluation of (7) requires sampling actions from π and trajectories from P_θ. However, because the flow model X_θ is trained on an offline dataset $\mathcal{D}$ with limited state-action

coverage, $P_\theta(\cdot \mid s,a)$ can be unreliable for unseen pairs. Sampling actions directly from π may therefore query the flow model on $(s,a) \notin \mathcal{D}$, causing extrapolation error of value estimates.

We estimate the value (7) without evaluating new actions out of the offline data by Inverse Propensity Weighting (IPW). Given dataset $\mathcal{D}$, we can reformulate the averaged objective (7) as

$$\mathbb{E}_{\tau\sim P_\theta(\cdot|s,a),a\sim\pi(\cdot|s),s\in\mathcal{D}}\big[E(\tau)\big] \approx \mathbb{E}_{\tau\sim P_\theta(\cdot|s,a),(s,a)\in\mathcal{D}}\big[w_\pi E(\tau)\big], \tag{8}$$

where $E(\tau) = \sum_{h=0}^{H-1}\gamma^h r_h + \gamma^H V_\psi^\pi(s_H)$ is the value expansion of reward-state trajectory τ and $w_\pi = \frac{\pi(a|s)}{\pi_0(a|s)}$. Here, we use the pretrained policy π_0 in Line 1 as a surrogate of the behavior policy π_b. If we learn a deterministic policy π based on Q_ϕ^π, we approximate the weight as $w_\pi = \frac{1}{\pi_0(a|s)}\mathrm{softmax}\left(\beta\cdot Q_\phi^\pi(s,a)\right)$ with constant $\beta>0$ using Q_ϕ^π fitted in the previous iteration.

The objective in (8) enables policy evaluation using the offline dataset. However, a low value does not necessarily correspond to a plausible adversarial trajectory generator, since different state-action pairs may have different margins for value degradation: some rewards may already be near their plausible lowest, while others may still have substantial room to decrease. Directly minimizing (8) may therefore push some trajectory kernels to generate unrealistically low values. To generate plausible worst-case returns, we define a relative value objective by comparing against a reference as

$$\mathbb{E}_{\tau\sim P_\theta(\cdot|s,a),(s,a)\in\mathcal{D}}\big[w_\pi \Delta_E(\tau)\big] = \mathbb{E}_{\tau\sim P_\theta(\cdot|s,a),(s,a)\in\mathcal{D}}\big[w_\pi\cdot(E(\tau) - \mathbb{E}_{\mathcal{D}}\left[E(\tau_{\mathcal{D}}(s,a))\right])\big], \tag{9}$$

where the reference $\mathbb{E}_{\mathcal{D}}\left[E(\tau_{\mathcal{D}}(s,a))\right]$ is the average reward over the trajectory $\tau_{\mathcal{D}}(s,a)$ drawn from $\mathcal{D}$ with initial (s,a). The relative value objective facilitates adversarial fine-tuning of the flow model so that its value objective can be evaluated more accurately for each state-action pair.

Tractable Reformulation Given the relative value objective (9) and the generative uncertainty set (5), Line 3 of GURO-RL can search the worst-case trajectory kernel with the objective

$$\min_\theta \mathbb{E}_{\tau\sim P_\theta(\cdot|s,a),(s,a)\in\mathcal{D}}\big[w_\pi\Delta_E(\tau)\big],\ \text{s.t. } J_{\mathrm{GFM}}(\theta)\le\epsilon. \tag{10}$$

While the constraint is the tractable guided flow-matching loss $J_{\mathrm{GFM}}(\theta)$, the objective in (10) depends on the flow-model parameter θ through its implicit distribution P_θ, so it can not be directly optimized. We therefore reformulate (10) as an optimization objective over θ using flow policy optimization (McAllister et al., 2025). Let θ_{old} denote the flow-model parameters from the previous iteration. The parameters θ are updated by transforming (10) into the PPO-style objective:

$$\max_\theta \mathbb{E}_{\tau\in P_{\theta_{\mathrm{old}}}(\cdot|s,a),(s,a)\in\mathcal{D}}\left[\min(r_\theta(\tau)A(\tau), \mathrm{clip}(r_\theta(\tau),1-\epsilon_{\mathrm{ppo}},1+\epsilon_{\mathrm{ppo}})\,A(\tau))\right] \tag{11a}$$

$$\text{s.t. } J_{\mathrm{GFM}}(\theta)\le\epsilon, \tag{11b}$$

where $A(\tau) = -w_\pi\Delta_E(\tau)$ is the relative objective in (9), $r_\theta(\tau) = \exp(\hat{J}_{\mathrm{GFM}}(\theta_{\mathrm{old}}) - \hat{J}_{\mathrm{GFM}}(\theta))$ approximates a scaled likelihood ratio between P_θ and $P_{\theta_{\mathrm{old}}}$ when $\hat{J}_{\mathrm{GFM}}(\theta) = \frac{1}{N_\epsilon}\sum_{i=1}^{N_\epsilon}\left[\left\|u_\theta^{gfm}(\tau_{k_i},k_i,(s,a)) - (\tau-\epsilon_i)\right\|_2^2\right]$ with $\epsilon_i \sim \mathcal{N}(0,\mathbf{I}), k_i \sim \mathrm{Uniform}([0,1])$ and $\tau_{k_i} = (1-k_i)\epsilon_i + k_i\tau$ (McAllister et al., 2025).

The value objective (7) is reformulated as the constrained optimization problem (11) over the flow-model parameters θ, which can be solved using standard constrained optimizers. In this work, we adopt a Lagrangian method. We introduce a nonnegative multiplier $\mu \ge 0$ and optimize $\theta \leftarrow \arg\max_\theta \left(L_{\mathrm{ppo}}(\theta,\mathcal{D}) - \mu\big(J_{\mathrm{FM}}(\theta;\mathcal{D}) - \epsilon\big)\right)$, where $L_{\mathrm{ppo}}(\theta,\mathcal{D})$ is the objective in (11a). Given the updated θ, the multiplier is updated by projected dual ascent: $\mu \leftarrow \left[\mu + \eta_\mu\big(J_{\mathrm{FM}}(\theta;\mathcal{D}) - \epsilon\big)\right]_+$, where $\eta_\mu > 0$ is the dual step size. This alternating update solves (11).

5.2.2 Conservative Policy Update

Given the fine-tuned flow model, Line 4 of GURO-RL samples a set $\mathcal{B}$ of plausible adversarial trajectories on which the previous policy performs poorly. Although the flow model can generate trajectories from arbitrary initial state-action pairs, the learned kernel $P_\theta(\cdot \mid s,a)$ may be unreliable for pairs not in the offline dataset, so we generate trajectories only from observed $(s,a)\in\mathcal{D}$, rather

than querying $P_\theta(\cdot \mid s, a)$ outside the offline data support. Thus, $\mathcal{B}$ remains supported by the offline data while capturing plausible shifts in rewards and transitions. Line 5 then applies conservative offline RL methods Kostrikov et al. (2021); Kumar et al. (2020); Fujimoto et al. (2019) to update the policy based on the augmented trajectories in $\mathcal{B}$. Instantiations of Line 5 with different offline RL backbones are provided in Section B. In this way, GURO−RL addresses distribution shift and limited offline coverage while exploiting generative uncertainty to improve policy robustness aware of diverse yet plausible environment shifts.

6 Experimental Evaluation

We evaluate GURO−RL along three dimensions: (i) **OOD robustness** (Appendix D.1): comparison with standard, robust, and generative-model baselines, including methods combining these paradigms, under OOD dynamics; (ii) **Design ablations** (Appendix D.2): evaluation of GURO−RL with different offline RL backbones for Line 5, assessment of IPW effectiveness, and sensitivity to the budget ϵ and rollout horizon H; (iii) **Domain application:** an EPANET-based Water-Distribution-Network (WDN) control case study assessing robustness to realistic dynamics shifts (Appendix E).

Datasets. For the default application, we use **Minari** MuJoCo datasets for *Hopper* (**H**), *HalfCheetah* (**HC**), and *Walker2d* (**W**) Younis et al. (2024). *Simple-Medium* (**SM**) and *Medium-Expert* (**ME**) mix the corresponding datasets equally at the trajectory level, preserving temporal consistency for multi-step targets. Task–dataset settings are abbreviated as, e.g., **H-SM**.

Baselines. We compare GURO−RL with the following baselines. **IQL** Kostrikov et al. (2021) is a standard offline RL method and the default backbone for GURO−RL. The backbone experiments additionally use **BCQ** Fujimoto et al. (2019) and **CQL** Kumar et al. (2020); all other experiments use IQL for GURO−RL. **ATLA** Zhang et al. (2021) alternates policy training with a learned adversary that perturbs state observations. **RFQI** Panaganti et al. (2022) learns from offline data using robust Bellman backups under total-variation transition uncertainty. **RSC** Ding et al. (2023) combines state interventions with a learned structural causal model to improve robustness against spurious correlations. **ROMB** Chen et al. (2026) jointly learns a robust policy and an adversarial world model through KL-constrained Stackelberg optimization. **FWM** is our flow-matching adaptation of DWM Ding et al. (2024), using non-adversarial multi-step rollouts for value learning. **PLR−PVL** Berdica et al. (2024) prioritizes replay across pretrained world models using positive value loss as a regret proxy, adapted here to a fixed flow ensemble. Implementation and adaptation details are provided in Appendix C.3.

Setup. For the default application, training consists of two stages: Stage-1 pretrains a shared backbone (Default IQL), and Stage 2 continues from its pretrained parameters for 25% of the Stage 1 update count, inserting one method-specific update after every 10 standard backbone updates to improve training efficiency. For each task, all applicable methods use the same number of training epochs with three random seeds. RFQI follows its native training procedure with robust Bellman updates rather than the two-stage protocol, while using the same dataset and evaluation protocol. For evaluation, we use **Robust Gymnasium** (Gu et al., 2025) to construct one nominal and 12 OOD environments per task by independently scaling morphology, friction, or gravity parameters by factors in $\{0.8, 0.9, 1.1, 1.2\}$. We report the mean and standard deviation of normalized scores across seeds. Detailed experimental settings are provided in Appendix C.4.

6.1 Main Results

Our main MuJoCo experiments compare GURO−RL with baselines under nominal dynamics and several types of OOD dynamics shifts. As shown in Table 2, GURO−RL achieves the highest average scores across nominal and shifted environments in all three **SM** settings, indicating that improved robustness of GURO−RL does not come at the expense of nominal performance in these setups. IQL is the offline RL backbone in Line 5 of GURO−RL. The improvement of GURO−RL relative to its IQL backbone demonstrates the effectiveness of generative trajectory augmentation and robust training.

Compared with FWM, offline RL with a flow-based world model, GURO−RL increases the mean OOD score from 54.4 to 58.0, with a particularly large gain under **H-SM** gravity shifts (72.0 versus 64.7).

Task	Scenario	Offline RL	World Model	Robust RL			Robust RL with Generative Model		
		IQL	FWM	ATLA	RFQI	RSC	ROMB	PLR-PVL	GURO-RL
H-SM	Nominal	84.3 ±4.8	87.7 ±5.0	82.2 ±7.7	83.6 ±3.4	89.1 ±2.4	87.6 ±5.8	87.2 ±4.4	89.9 ±2.4
	Morphology	49.2 ±0.8	48.4 ±4.6	49.8 ±4.5	52.5 ±1.0	49.8 ±2.7	51.6 ±2.2	48.3 ±8.3	48.1 ±7.9
	Friction	42.7 ±11.3	43.8 ±5.3	52.8 ±4.9	35.0 ±3.7	49.2 ±13.0	47.8 ±7.0	46.3 ±6.6	53.1 ±8.5
	Gravity	60.0 ±17.5	64.7 ±7.7	68.2 ±11.2	45.7 ±3.2	67.2 ±13.4	64.2 ±14.4	65.9 ±13.5	72.0 ±13.0
	Average	59.1	61.2	63.3	54.2	63.8	62.8	61.9	65.8
W-SM	Nominal	55.4 ±6.2	68.7 ±9.2	61.6 ±13.6	56.0 ±1.4	71.8 ±2.9	57.8 ±5.5	60.6 ±9.7	75.4 ±5.2
	Morphology	55.7 ±5.1	58.5 ±6.6	56.8 ±1.2	51.6 ±0.4	56.6 ±2.6	59.1 ±4.8	55.5 ±7.6	61.4 ±9.5
	Friction	57.5 ±5.7	64.0 ±6.6	61.6 ±9.8	56.8 ±0.9	67.0 ±4.6	60.9 ±5.7	60.9 ±9.5	65.0 ±8.9
	Gravity	45.0 ±1.2	45.5 ±3.9	44.4 ±0.8	53.1 ±0.8	45.3 ±2.1	44.7 ±2.3	43.6 ±1.8	50.1 ±1.8
	Average	53.4	59.2	56.1	54.4	60.2	55.6	55.1	63.0
HC-SM	Nominal	62.7 ±4.2	61.8 ±0.7	64.9 ±1.6	47.6 ±3.4	65.3 ±6.0	65.0 ±1.3	64.3 ±2.6	74.4 ±3.8
	Morphology	54.5 ±1.0	54.8 ±1.9	53.8 ±2.6	44.4 ±1.4	55.0 ±3.5	54.3 ±1.3	55.8 ±2.8	54.2 ±0.3
	Friction	63.7 ±2.4	65.1 ±2.4	61.0 ±0.3	46.0 ±2.1	64.0 ±4.1	64.3 ±2.4	63.7 ±5.3	70.2 ±4.3
	Gravity	43.9 ±2.6	44.6 ±2.8	43.4 ±2.0	43.0 ±0.7	46.7 ±3.4	44.8 ±3.6	45.7 ±1.7	47.7 ±1.3
	Average	56.2	56.6	55.8	45.2	57.8	57.1	57.4	61.6
Average	OOD	52.5	54.4	54.6	47.6	55.6	54.6	54.0	58.0
	ALL	56.2	59.0	58.4	51.3	60.6	58.5	58.1	63.5

Table 2: Partial MuJoCo performance comparison. Black shades denote OOD scenarios and blue shades denote the best performance. ALL denotes the average score of all scenarios.

Since both methods use conditional flow models for multi-step generation, this comparison suggests the benefit from robust training with generative uncertainty set: GURO-RL actively iteratively seeks adverse trajectory kernels to learn worst-case critic for robustness, while the flow-matching constraint limits discrepancy of the kernels from the training distribution for nominal performance.

GURO-RL demonstrates superiority compared to robust RL baselines without using generative models. Relative to RSC and ATLA, GURO-RL improves the scores in both OOD and nominal scenarios. These demonstrate the effectiveness of the generative uncertainty set in (5) for capturing diverse yet plausible distribution shifts by implicit generative model, improving robustness without inducing overly pessimistic policies. The advantage over RFQI is larger (12.2 in all average). This may come from non-rectangularity of the generative uncertainty set (5) which restricts arbitrary combinations of local perturbations for plausible environment shift, while the independent local worst-case value evaluation in RFQI may be overly pessimistic when physical changes couple transitions.

Among baselines using generative models, GURO-RL outperforms both ROMB and PLR-PVL under both nominal and OOD settings. This demonstrates the stronger capability of the generative uncertainty set to model plausible distributions in high-dimensional continuous spaces, compared with the explicit probabilistic uncertainty set in ROMB and the finite model set in PLR-PVL. The results also validate the adversarial flow-model fine-tuning (Line 3 in GURO-RL), showing that the reformulated objective can effectively identify worst-case trajectory distributions.

The complete main results and analysis are provided in Appendix D.1.

7 Conclusion

We introduced GURO-RL, a robust offline RL framework that leverages flow models to construct a generative uncertainty set over environment dynamics. We introduce a generative uncertainty set to capture non-rectangular, diverse yet plausible distribution shifts, while a tractable discrepancy metric based on flow-matching loss maintains consistency of the candidate distributions with the offline data. We reformulate the value objective based on the offline dataset so that GURO-RL can adversarially fine-tune the flow model within the generative uncertainty set and incorporate the generated adverse trajectories into conservative value-based policy learning. Experiments on MuJoCo and an EPANET-based application demonstrate improved policy robustness across diverse nominal and OOD conditions. *Broader impact:* This work highlights the superiority of generative uncertainty modeling for robust offline RL and has the potential to improve robustness for physical control problems like robotics, industrial control, and infrastructure systems.

AI USE STATEMENT

In this work, we used generative AI tools to assist with manuscript drafting and revision, including improving language, clarity, organization, and presentation. In addition, we use AI tools to check the correctness of proofs in Appendix A. We also used generative AI tools to generate synthetic water-demand datasets for the water-distribution-network case study. These synthetic datasets were used exclusively to construct demand-shift out-of-distribution (OOD) evaluation scenarios.

All AI-assisted manuscript content was reviewed and revised by the authors for technical accuracy, consistency, and appropriate attribution. The AI-generated synthetic data were also reviewed to ensure consistency with the intended experimental settings and were used only within the evaluation protocol described in the paper. We take responsibility for the final content of this paper, including all text, claims, and experimental results.

ETHICS STATEMENT

This work focuses on algorithmic development and evaluation in simulated environments and does not involve human subjects, personally identifiable information, or sensitive real-world data. The proposed methods are intended for methodological research in robust offline reinforcement learning, and we are not aware of specific ethical concerns arising from the contents presented in this paper.

REPRODUCIBILITY STATEMENT

We have made every effort to ensure reproducibility. The complete procedure of `GURO-RL` is provided in Algorithm 1 with concrete instantiations in Algorithms 2, 3, 4 in Appendix B. The assumptions and complete proof of the Wasserstein consistency bound of Proposition 4.1 are provided in Appendix A. Appendix C reports the MuJoCo experimental setup, including dataset construction, environment-shift settings, model architectures, hyperparameters, baseline configurations, and evaluation protocols. Appendix E provides the corresponding formulation, dataset, environment configuration, training setup, and evaluation procedure for the water-system case study.

REFERENCES

Mohammed Amin Abdullah, Hang Ren, Haitham Bou Ammar, Vladimir Milenkovic, Rui Luo, Mingtian Zhang, and Jun Wang. Wasserstein Robust Reinforcement Learning. *arXiv preprint arXiv:1907.13196*, 2019. URL https://arxiv.org/abs/1907.13196.

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is Conditional Generative Modeling all you need for Decision-Making? arXiv preprint arXiv:2211.15657, 2023. URL https://arxiv.org/abs/2211.15657.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for World Modeling: Visual Details Matter in Atari. In *Advances in Neural Information Processing Systems*, volume 37, pp. 58757–58791. Curran Associates, Inc., 2024. doi: 10.52202/079017-1873. URL https://doi.org/10.52202/079017-1873.

Gissella Bejarano, Adita Kulkarni, Raushan Raushan, Anand Seetharam, and Arti Ramesh. SWaP: Probabilistic Graphical and Deep Learning Models for Water Consumption Prediction. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, BuildSys '19, pp. 233–242. ACM, 2019. doi: 10.1145/3360322.3360846. URL https://doi.org/10.1145/3360322.3360846.

Uljad Berdica, Kelvin Li, Michael Beukman, Alexander David Goldie, Perla Maiolino, and Jakob Nicolaus Foerster. Robust Offline Learning via Adversarial World Models. In *NeurIPS 2024 Workshop on Open-World Agents*, 2024. URL https://openreview.net/forum?id=dwKIyvYURI.

Jose Blanchet, Miao Lu, Tong Zhang, and Han Zhong. Double Pessimism is Provably Efficient for Distributionally Robust Offline Reinforcement Learning: Generic Algorithm and Robust Partial Coverage. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://papers.neurips.cc/paper_files/paper/2023/hash/d31b005d817e9c635ec8ffb0fb90190e-Abstract-Conference.html.

Jiayu Chen, Le Xu, Aravind Venugopal, and Jeff Schneider. Policy-Driven World Model Adaptation for Robust Offline Model-based Reinforcement Learning. arXiv preprint arXiv:2505.13709, 2026. URL https://arxiv.org/abs/2505.13709.

Xin Chen, Guannan Qu, Yujie Tang, Steven Low, and Na Li. Reinforcement learning for selective key applications in power systems: Recent advances and future challenges. *IEEE Transactions on Smart Grid*, 13(4):2935–2958, 2022.

Zijun Chen, Shengbo Wang, and Nian Si. Sample Complexity of Distributionally Robust Average-Reward Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 85402–85463. Curran Associates, Inc., 2025. doi: 10.52202/085713-2862. URL https://doi.org/10.52202/085713-2862.

Pierre Clavier, Laixi Shi, Erwan Le Pennec, Eric Mazumdar, Adam Wierman, and Matthieu Geist. Near-Optimal Distributionally Robust Reinforcement Learning with General L_p Norms. In *Advances in Neural Information Processing Systems*, volume 37, pp. 1750–1810. Curran Associates, Inc., 2024. doi: 10.52202/079017-0056. URL https://doi.org/10.52202/079017-0056.

Anthony Coache and Sebastian Jaimungal. Robust Reinforcement Learning with Dynamic Distortion Risk Measures. *SIAM Journal on Mathematics of Data Science*, 8(1):1–22, 2026. doi: 10.1137/24m1699802. URL https://doi.org/10.1137/24m1699802.

Tuan Quang Dam, Kishan Panaganti, Brahim Driss, and Adam Wierman. Online Robust Reinforcement Learning Through Monte-Carlo Planning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 12314–12343, 2025. URL https://proceedings.mlr.press/v267/dam25a.html.

Wenhao Ding, Laixi Shi, Yuejie Chi, and Ding Zhao. Seeing is not Believing: Robust Reinforcement Learning against Spurious Correlation. In *Advances in Neural Information Processing Systems*, volume 36, pp. 66328–66363. Curran Associates, Inc., 2023. doi: 10.52202/075280-2895. URL https://doi.org/10.52202/075280-2895.

Zihan Ding, Amy Zhang, Yuandong Tian, and Qinqing Zheng. Diffusion World Model: Future Modeling Beyond Step-by-Step Rollout for Offline Reinforcement Learning. arXiv preprint arXiv:2402.03570, 2024. URL https://arxiv.org/abs/2402.03570.

Daniele Foffano, Alessio Russo, and Alexandre Proutiere. Adversarial Diffusion for Robust Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 68627–68658. Curran Associates, Inc., 2025. doi: 10.52202/085713-2309. URL https://doi.org/10.52202/085713-2309.

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

Scott Fujimoto, David Meger, and Doina Precup. Off-Policy Deep Reinforcement Learning without Exploration. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2052–2062. PMLR, 2019. URL https://proceedings.mlr.press/v97/fujimoto19a.html.

Shangding Gu, Laixi Shi, Muning Wen, Ming Jin, Eric Mazumdar, Yuejie Chi, Adam Wierman, and Costas Spanos. Robust Gymnasium: A Unified Modular Benchmark for Robust Reinforcement Learning. arXiv preprint arXiv:2502.19652, 2025. URL https://arxiv.org/abs/2502.19652.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *CoRR*, abs/1801.01290, 2018. URL http://arxiv.org/abs/1801.01290.

Zhaolin Hu and L Jeff Hong. Kullback-leibler divergence constrained distributionally robust optimization. *Available at Optimization Online*, 1(2):9, 2013.

Garud N. Iyengar. Robust Dynamic Programming. *Mathematics of Operations Research*, 30(2): 257–280, 2005. doi: 10.1287/moor.1040.0129. URL https://doi.org/10.1287/moor.1040.0129.

Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with Diffusion for Flexible Behavior Synthesis. arXiv preprint arXiv:2205.09991, 2022. URL https://arxiv.org/abs/2205.09991.

Zhennan Jiang, Kai Liu, Yuxing Qin, Shuai Tian, Yupeng Zheng, Mingcai Zhou, Zhengtao Zhang, Chao Yu, Haoran Li, and Dongbin Zhao. World4rl: Diffusion world models for policy refinement with reinforcement learning for robotic manipulation. *IEEE Robotics and Automation Letters*, 2026.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline Reinforcement Learning with Implicit Q-Learning. arXiv preprint arXiv:2110.06169, 2021. URL https://arxiv.org/abs/2110.06169.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative Q-Learning for Offline Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 1179–1191. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/0d2b2061826a5df3221116a5085a6052-Paper.pdf.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems. arXiv preprint arXiv:2005.01643, 2020. URL https://arxiv.org/abs/2005.01643.

Mengmeng Li, Daniel Kuhn, and Tobias Sutter. Towards optimal offline reinforcement learning. *arXiv preprint arXiv:2503.12283*, 2025.

Mengmeng Li, Daniel Kuhn, and Tobias Sutter. Policy gradient algorithms for robust mdp with nonrectangular uncertainty sets. *SIAM Journal on Optimization*, 36(1):120–151, 2026.

Yan Li and Alexander Shapiro. Rectangularity and Duality of Distributionally Robust Markov Decision Processes. *Mathematical Programming*, 2025. doi: 10.1007/s10107-025-02297-y. URL https://doi.org/10.1007/s10107-025-02297-y.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for Generative Modeling. arXiv preprint arXiv:2210.02747, 2023. URL https://arxiv.org/abs/2210.02747.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky T. Q. Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow Matching Guide and Code. arXiv preprint arXiv:2412.06264, 2024. URL https://arxiv.org/abs/2412.06264.

Zhishuai Liu and Pan Xu. Minimax optimal and computationally efficient algorithms for distributionally robust offline reinforcement learning. *Advances in Neural Information Processing Systems*, 37:86602–86654, 2024.

Zijian Liu, Qinxun Bai, Jose Blanchet, Perry Dong, Wei Xu, Zhengqing Zhou, and Zhengyuan Zhou. Distributionally robust q-learning. In *International Conference on Machine Learning*, pp. 13623–13643. PMLR, 2022.

Miao Lu, Han Zhong, Tong Zhang, and Jose Blanchet. Distributionally robust reinforcement learning with interactive data collection: Fundamental hardness and near-optimal algorithms. *Advances in Neural Information Processing Systems*, 37:12528–12580, 2024.

Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*, 2022.

Xiaoteng Ma, Zhipeng Liang, Jose Blanchet, Mingwen Liu, Li Xia, Jiheng Zhang, Qianchuan Zhao, and Zhengyuan Zhou. Distributionally robust offline reinforcement learning with linear function approximation. *arXiv preprint arXiv:2209.06620*, 2022.

David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow Matching Policy Gradients. arXiv preprint arXiv:2507.21053, 2025. URL https://arxiv.org/abs/2507.21053.

Peyman Mohajerin Esfahani and Daniel Kuhn. Data-driven distributionally robust optimization using the Wasserstein metric: performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1-2):115–166, 2018. doi: 10.1007/s10107-017-1172-1. URL https://doi.org/10.1007/s10107-017-1172-1.

Fabio Muratore, Fabio Ramos, Greg Turk, Wenhao Yu, Michael Gienger, and Jan Peters. Robot learning from randomized simulations: A review. *Frontiers in Robotics and AI*, 9:799893, 2022.

Arnab Nilim and Laurent El Ghaoui. Robustness in Markov Decision Problems with Uncertain Transition Matrices. In *Advances in Neural Information Processing Systems*, volume 16. MIT Press, 2003. URL https://proceedings.neurips.cc/paper_files/paper/2003/file/300891a62162b960cf02ce3827bb363c-Paper.pdf.

Kishan Panaganti, Zaiyan Xu, Dileep Kalathil, and Mohammad Ghavamzadeh. Robust Reinforcement Learning using Offline Data. In *Advances in Neural Information Processing Systems*, volume 35, pp. 32211–32224. Curran Associates, Inc., 2022. doi: 10.52202/068431-2334. URL https://doi.org/10.52202/068431-2334.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3803–3810. IEEE, 2018. doi: 10.1109/icra.2018.8460528. URL https://doi.org/10.1109/icra.2018.8460528.

Rafael Figueiredo Prudencio, Marcos ROA Maximo, and Esther Luna Colombini. A survey on offline reinforcement learning: Taxonomy, review, and open problems. *IEEE transactions on neural networks and learning systems*, 35(8):10237–10257, 2023.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, 1994. ISBN 9780470316887. doi: 10.1002/9780470316887. URL https://doi.org/10.1002/9780470316887.

Anirudh Satheesh, Ziyi Chen, Furong Huang, and Heng Huang. Provably efficient algorithms for s-and non-rectangular robust mdps with general parameterization. *arXiv preprint arXiv:2602.11387*, 2026.

Laixi Shi and Yuejie Chi. Distributionally Robust Model-Based Offline Reinforcement Learning with Near-Optimal Sample Complexity. *Journal of Machine Learning Research*, 25(200):1–91, 2024. URL http://jmlr.org/papers/v25/22-1482.html.

Elena Smirnova, Elvis Dohmatob, and Jérémie Mary. Distributionally Robust Reinforcement Learning. arXiv preprint arXiv:1902.08708, 2019. URL https://arxiv.org/abs/1902.08708.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 2 edition, 2018. ISBN 9780262039246. URL https://mitpress.mit.edu/9780262039246/reinforcement-learning/.

He Wang, Laixi Shi, and Yuejie Chi. Sample Complexity of Offline Distributionally Robust Linear Markov Decision Processes. arXiv preprint arXiv:2403.12946, 2024. URL https://arxiv.org/abs/2403.12946.

Qiuhao Wang, Yuqi Zha, Chin Pang Ho, and Marek Petrik. Provable policy gradient for robust average-reward mdps beyond rectangularity. In *Forty-second International Conference on Machine Learning*, 2025.

Shengbo Wang and Nian Si. Non-Rectangular Average-Reward Robust MDPs: Optimal Policies and Their Transient Values. arXiv preprint arXiv:2603.00945, 2026. URL https://arxiv.org/abs/2603.00945.

Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. Robust Markov Decision Processes. *Mathematics of Operations Research*, 38(1):153–183, 2013. doi: 10.1287/moor.1120.0566. URL https://doi.org/10.1287/moor.1120.0566.

Huan Xu and Shie Mannor. Distributionally Robust Markov Decision Processes. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010. URL https://proceedings.neurips.cc/paper_files/paper/2010/file/19f3cd308f1455b3fa09a282e0d496f4-Paper.pdf.

Yang Xu, Swetha Ganesh, and Vaneet Aggarwal. Efficient Q-Learning and Actor–Critic Methods for Robust Average-Reward Reinforcement Learning. In *Proceedings of the 42nd Conference on Uncertainty in Artificial Intelligence*, volume 337, pp. 7613–7650, 2026. URL https://proceedings.mlr.press/v337/xu26d.html.

Zaiyan Xu, Kishan Panaganti, and Dileep Kalathil. Improved sample complexity bounds for distributionally robust reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 9728–9754. PMLR, 2023.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Omar G. Younis, Rodrigo Perez-Vicente, John U. Balis, Will Dudley, Alex Davey, and Jordan K Terry. Minari, 2024. URL https://doi.org/10.5281/zenodo.13767625. Software, version 0.5.0.

Chi Zhang, Zain Ulabedeen Farhat, George Atia, and Yue Wang. Model-free offline reinforcement learning with enhanced robustness. In *International Conference on Learning Representations*, volume 2025, pp. 61209–61253, 2025.

Huan Zhang, Hongge Chen, Duane Boning, and Cho-Jui Hsieh. Robust Reinforcement Learning on State Observations with Learned Optimal Adversary. arXiv preprint arXiv:2101.08452, 2021. URL https://arxiv.org/abs/2101.08452.

Qinqing Zheng, Matt Le, Neta Shaul, Yaron Lipman, Aditya Grover, and Ricky T. Q. Chen. Guided Flows for Generative Modeling and Decision Making. arXiv preprint arXiv:2311.13443, 2023. URL https://arxiv.org/abs/2311.13443.

Zhengqing Zhou, Zhengyuan Zhou, Qinxun Bai, Linhai Qiu, Jose Blanchet, and Peter Glynn. Finite-sample regret bound for distributionally robust offline tabular reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 3331–3339. PMLR, 2021.

A PROOF OF PROPOSITION 4.1

Assumptions.

(A1) $\mathbb{E}[\|X_0\|_2^2 + \|X_1\|_2^2] < \infty$.

(A2) There is a measurable $\mathsf{Y}_0 \subseteq \mathsf{Y}$, with $\nu(\mathsf{Y}_0) = 1$, on which the ODEs $\dot{z}_k = u^\star(z_k, k, y)$ and $\dot{z}_k = u_\theta^{\mathrm{gfm}}(z_k, k, y)$ admit jointly Borel flow maps $\Phi_{k,y}^\star$ and $\Phi_{k,y}^{\theta,\mathrm{gfm}}$. For every $(y, x) \in \mathsf{Y}_0 \times \mathsf{X}$, both trajectories are absolutely continuous, start at $\Phi_{0,y}^\star(x) = \Phi_{0,y}^{\theta,\mathrm{gfm}}(x) = x$, satisfy their ODEs almost everywhere in k, and the ideal flow obeys $(\Phi_{k,y}^\star)_\# P_0 = P_k(\cdot \mid y)$ for all $(k, y) \in [0, 1] \times \mathsf{Y}_0$.

(A3) For some finite $L_{\mathrm{gfm}} \geq 0$, uniformly over $(k, y) \in [0, 1] \times \mathsf{Y}_0$, $\|u_\theta^{\mathrm{gfm}}(x, k, y) - u_\theta^{\mathrm{gfm}}(x', k, y)\|_2 \leq L_{\mathrm{gfm}}\|x - x'\|_2$ for all $x, x' \in \mathsf{X}$.

(A4) $J_{\mathrm{GFM}}(\theta) < \infty$.

Proof. Let $U = X_1 - X_0$ be the difference between the source and target random variables X_0 and X_1. Since the condition $\widetilde{Y}$ is drawn from $\{\varnothing, Y\}$ with a Bernoulli probability of p_{drop}, the training loss of guided flow matching can be expressed as

$$\begin{aligned} J_{\mathrm{GFM}}(\theta) =& \mathbb{E}\left[\left\| u_\theta(X_k, k, \widetilde{Y}) - (X_1 - X_0)\right\|_2^2\right] \\ =& (1 - p_{\mathrm{drop}})\mathbb{E}\|u_\theta(X_k, k, Y) - U\|_2^2 + p_{\mathrm{drop}}\mathbb{E}\|u_\theta(X_K, K, \varnothing) - U\|_2^2. \end{aligned} \tag{12}$$

Since the sampling guided vector field is expressed as $u_\theta^{\mathrm{gfm}}(x, k, y) = \lambda_{\mathrm{gfm}} u_\theta(x, k, y) + (1 - \lambda_{\mathrm{gfm}})u_\theta(x, k, \varnothing), \lambda_{\mathrm{gfm}} \in (0, 1)$, the sampling guided flow matching loss is bounded by weighted Cauchy–Schwarz as

$$\mathbb{E}\left[\left\| u_\theta^{\mathrm{gfm}}(X_k, k, Y) - U\right\|_2^2\right] \leq C_{\mathrm{gfm}} J_{\mathrm{GFM}}(\theta), \tag{13}$$

where $C_{\mathrm{gfm}} = \lambda_{\mathrm{gfm}}^2/(1 - p_{\mathrm{drop}}) + (1 - \lambda_{\mathrm{gfm}})^2/p_{\mathrm{drop}}$.

Denote $u^*(x, k, y) = \mathbb{E}[U \mid X_k = x,\ Y = y]$ as the population-optimal vector field. Let $\mathcal{G} = \sigma(X_k, k, Y)$ be the sigma-algebra generated by the random variables (X_k, k, Y). We have $u^*(X_k, k, Y) = \mathbb{E}[U \mid \mathcal{G}]$. Assumption (A1) gives $U \in L^2$; together with Eq. (13), this also gives $u_\theta^{\mathrm{gfm}}(X_k, k, Y) \in L^2$. Thus, we have

$$\mathbb{E}\|U - u_\theta^{\mathrm{gfm}}(X_k, k, Y)\|_2^2 = \mathbb{E}\|U - u^*(X_k, k, Y)\|_2^2 + \mathbb{E}\|u^*(X_k, k, Y) - u_\theta^{\mathrm{gfm}}(X_k, k, Y)\|_2^2,$$

which holds because $\mathbb{E}\left[\langle U - u^*(X_k, k, Y), u^*(X_k, k, Y) - u_\theta^{\mathrm{gfm}}(X_k, k, Y)\rangle\right] = \mathbb{E}\left[\left\langle \mathbb{E}[U - u^*(X_k, k, Y) \mid \mathcal{G}], u^*(X_k, k, Y) - u_\theta^{\mathrm{gfm}}(X_k, k, Y)\right\rangle\right] = 0$. Thus, it holds that

$$\mathbb{E}[\|u^*(X_k, k, Y) - u_\theta^{\mathrm{gfm}}(X_k, k, Y)\|_2^2] \leq \mathbb{E}[\|u_\theta^{\mathrm{gfm}}(X_k, k, Y) - U\|_2^2] \leq C_{\mathrm{gfm}} J_{\mathrm{GFM}}(\theta). \tag{14}$$

Let $X_k^* = \Phi_{k,y}^\star(X_0)$ be the point reached at flow time k following the optimal vector field u^* and $X_k^\theta = \Phi_{k,y}^{\theta,\mathrm{gfm}}(X_0)$ be the point reached at flow time k following u_θ^{gfm}. Their difference is $e_k = X_k^* - X_k^\theta$. Denote $\delta_k = u^*(X_k^*, k, y) - u_\theta^{\mathrm{gfm}}(X_k^*, k, y)$. The derivative of e_k with respect to k is expressed as

$$\dot{e}_k = \dot{X}_k^\star - \dot{X}_k^\theta = u^\star(X_k^\star, k, y) - u_\theta^{\mathrm{gfm}}(X_k^\theta, k, y) = \delta_k + u_\theta^{\mathrm{gfm}}(X_k^*, k, y) - u_\theta^{\mathrm{gfm}}(X_k^*, k, y). \tag{15}$$

Since $e_0 = 0$, integrating Eqn. (15) from 0 to k, we have

$$e_k = \int_0^k \delta_{k'}\, dk' + \int_0^k \left[u_\theta^{\mathrm{gfm}}(X_{k'}^*, k', y) - u_\theta^{\mathrm{gfm}}(X_{k'}^\theta, k', y)\right] dk'.$$

By triangle inequality, it holds that

$$\begin{aligned} \|e_k\|_2 &\leq \int_0^k \|\delta_{k'}\|_2\, dk' + \int_0^k \left\| u_\theta^{\mathrm{gfm}}(X_{k'}^*, k', y) - u_\theta^{\mathrm{gfm}}(X_{k'}^\theta, k', y)\right\|_2 dk' \\ &\leq \int_0^k \|\delta_{k'}\|_2\, dk' + L_{\mathrm{gfm}} \int_0^k \|e_{k'}\|_2 dk', \end{aligned}$$

where the last inequality holds by $L_{\mathrm{gfm}}-$Lipschitz of u_θ^{gfm} in Assumption (A3). By the integral Gronwall inequality, we have

$$\|e_k\|_2 \leq \int_0^k e^{L_{\mathrm{gfm}}(k-k')} \|\delta_{k'}\|_2 dk'.$$

At the end point of the ODE, it holds that

$$\begin{aligned} \|e_1\|_2^2 &\leq \left(\int_0^1 e^{L_{\mathrm{gfm}}(1-k')} \|\delta_{k'}\|_2 \, dk' \right)^2 \\ &\leq \left(\int_0^1 e^{2L_{\mathrm{gfm}}(1-k')} \, dk' \right) \left(\int_0^1 \|\delta_{k'}\|_2^2 \, dk' \right) \\ &\leq e^{2L_{\mathrm{gfm}}} \left(\int_0^1 \|u^*(X_k^*, k, y) - u_\theta^{\mathrm{gfm}}(X_k^*, k, y)\|_2^2 \, dk' \right), \end{aligned} \tag{16}$$

where the second inequality holds by Cauchy-Schwarz inequality.

It holds that $\mathbb{E}_{X_0 \sim P_0} \|e_1\|_2^2 \leq e^{2L_{\mathrm{cfg}}} \int_0^1 \mathbb{E}_{X_0 \sim P_0} \|\delta_k\|_2^2 \, dk < \infty$, and $\mathbb{E}_{X_0 \sim P_0} \|X_1^\theta\|_2^2 \leq 2\mathbb{E}_{X_0 \sim P_0} \|X_1^*\|_2^2 + 2\mathbb{E}_{X_0 \sim P_0} \|e_1\|_2^2 < \infty$. Thus, we can bound the squared Wasserstein distance between $P_\theta(\cdot \mid y)$ and $P_1(\cdot \mid y)$ for any condition y as

$$W_2^2(P_1(\cdot \mid y), P_\theta(\cdot \mid y)) \leq \mathbb{E}_{X_0 \sim P_0}[\|X_1^* - X_1^\theta\|_2^2] = E_{X_0 \sim P_0}[\|e_1\|_2^2].$$

Taking expectation over condition Y, we have

$$\begin{aligned} \mathbb{E}_Y[W_2^2(P_1(\cdot \mid Y), P_\theta(\cdot \mid Y))] &\leq \mathbb{E}_Y[\mathbb{E}_{X_0 \sim P_0}[\|e_1\|_2^2]] \\ &\leq e^{2L_{\mathrm{gfm}}} E_Y[\int_0^1 \mathbb{E}_{X_0 \sim P_0} \left[\|u^*(X_k^*, k, y) - u_\theta^{\mathrm{gfm}}(X_k^*, k, y)\|_2^2 \right] dk] \\ &\leq e^{2L_{\mathrm{gfm}}} C_{\mathrm{gfm}} J_{\mathrm{GFM}}(\theta), \end{aligned}$$

where the first inequality holds by the definition of Wasserstein distance, the second inequality holds by (16), and the last inequality holds by (14). □

B POLICY UPDATE METHODS IN GURO−RL

We follow the main-text notation: θ parameterizes the flow model, Q_ϕ^π and V_ψ^π are the learned value functions, and $Q_{\bar{\phi}}^\pi$ is the target critic. For IQL and CQL, π denotes π_{ϑ_π}, with actor parameters ϑ_π. We retain $E(\tau)$ for the bootstrapped trajectory return, with each backbone's terminal bootstrap written explicitly below. In all three instantiations, $\mathcal{B}$ denotes the adversarial batch from Line 4 of Algorithm 1.

B.1 IQL INSTANTIATIONS OF GURO−RL

Algorithm 2 instantiates Line 5 of Algorithm 1 with the IQL update of Kostrikov et al. (2021). Lines 1–4 of Algorithm 1 are unchanged. In particular, Line 4 produces the adversarial batch $\mathcal{B}$ from the fine-tuned flow model. Because each generated trajectory is conditioned on a state–action pair from the offline dataset, we write this batch as

$$\mathcal{B} = \left\{ \left(y_{\mathcal{D},i}, \tau_{\theta,i}^{\mathrm{adv}} \right) \right\}_{i=1}^{B}, \qquad y_{\mathcal{D},i} = (s_{0,i}, a_{0,i}).$$

Thus, the only backbone-specific operation in Line 5 is to apply IQL to this batch: the real initial context $y_{\mathcal{D},i}$ is used by IQL's value and policy updates, while the generated continuation $\tau_{\theta,i}^{\mathrm{adv}}$ replaces the standard one-step critic target by an adversarial H-step target.

For IQL, the terminal bootstrap is given directly by its value network V_ψ^π. For $\tau_\theta^{\mathrm{adv}} = (r_0^{\mathrm{adv}}, s_1^{\mathrm{adv}}, \ldots, r_{H-1}^{\mathrm{adv}}, s_H^{\mathrm{adv}})$, define the adversarial H-step target as

$$E(\tau_\theta^{\mathrm{adv}}) = \sum_{h=0}^{H-1} \gamma^h r_h^{\mathrm{adv}} + \gamma^H V_\psi^\pi(s_H^{\mathrm{adv}}). \tag{17}$$

For trajectories that terminate before H, rewards after termination and the terminal bootstrap are zero. With expectile level $\alpha_{\mathrm{IQL}} \in (0,1)$, define $\ell_{\alpha_{\mathrm{IQL}}}(u) = |\alpha_{\mathrm{IQL}} - \mathbb{1}\{u < 0\}|u^2$. Let $\beta_{\mathrm{IQL}} > 0$ be the advantage-weight inverse temperature, distinct from the IPW inverse temperature β in the main text. The IQL losses used in Line 5 are

$$\mathcal{L}_V(\psi; \mathcal{B}) = \mathbb{E}_{\mathcal{B}} \left[\ell_{\alpha_{\mathrm{IQL}}} \left(Q_{\bar{\phi}}^\pi(y_{\mathcal{D}}) - V_\psi^\pi(s_0) \right) \right], \tag{18}$$

$$\mathcal{L}_Q^{\mathrm{rob}}(\phi; \mathcal{B}) = \mathbb{E}_{\mathcal{B}} \left[\left(Q_\phi^\pi(y_{\mathcal{D}}) - E(\tau_\theta^{\mathrm{adv}}) \right)^2 \right], \tag{19}$$

$$\mathcal{L}_\pi(\vartheta_\pi; \mathcal{B}) = -\mathbb{E}_{\mathcal{B}} \left[e^{\beta_{\mathrm{IQL}}[Q_{\bar{\phi}}^\pi(y_{\mathcal{D}}) - V_\psi^\pi(s_0)]} \log \pi(a_0 \mid s_0) \right]. \tag{20}$$

Here $\mathbb{E}_{\mathcal{B}}$ denotes the empirical average over the batch produced in Line 4. The value loss retains IQL's expectile regression, and the policy loss retains its advantage-weighted behavior cloning on the real initial state–action pairs contained in $\mathcal{B}$. GURO−RL changes only the critic regression target: instead of IQL's standard one-step data target, the critic is regressed toward the adversarial H-step target in Eq. (17). During the Line 5 update, the generated trajectories in $\mathcal{B}$ and their H-step targets are treated as fixed samples; each loss differentiates only through its corresponding IQL network.

Algorithm 2 IQL based GURO−RL:Line 5 Update

Require: Adversarial batch $\mathcal{B} = \{(y_{\mathcal{D},i}, \tau_{\theta,i}^{\mathrm{adv}})\}_{i=1}^{B}$ from Line 4 of Algorithm 1; current IQL parameters $\psi, \phi, \bar{\phi}, \vartheta_\pi$; learning rates η_V, η_Q, η_π; target-update rate ρ

1: Form the fixed robust target $Q_H(\tau_{\theta,i}^{\mathrm{adv}}; V_\psi)$ for each sample using Eq. (17)
2: $\phi \leftarrow \phi - \eta_Q \nabla_\phi \mathcal{L}_Q^{\mathrm{rob}}(\phi; \mathcal{B})$
3: $\psi \leftarrow \psi - \eta_V \nabla_\psi \mathcal{L}_V(\psi; \mathcal{B})$
4: $\vartheta_\pi \leftarrow \vartheta_\pi - \eta_\pi \nabla_{\vartheta_\pi} \mathcal{L}_\pi(\vartheta_\pi; \mathcal{B})$
5: $\bar{\phi} \leftarrow (1-\rho)\bar{\phi} + \rho\phi$
6: **return** updated $\psi, \phi, \bar{\phi}, \vartheta_\pi$

B.2 CQL INSTANTIATIONS OF GURO−RL

Algorithm 3 instantiates Line 5 of Algorithm 1 with the actor–critic version of CQL (Kumar et al., 2020). Lines 1–4 are unchanged and supply the same adversarial batch $\mathcal{B} = \{(y_{\mathcal{D},i}, \tau^{\mathrm{adv}}_{\theta,i})\}_{i=1}^{B}$, where $y_{\mathcal{D},i} = (s_{0,i}, a_{0,i})$. The real initial contexts support CQL's conservative regularizer and policy update; the generated continuations supply the critic targets.

CQL retains policy $\pi = \pi_{\vartheta_\pi}$, critic Q^π_ϕ, and target critic $Q^\pi_{\bar\phi}$. Its terminal bootstrap uses the entropy-adjusted target critic under π, without a separate value network. For each supplied trajectory, the adversarial H-step target is

$$E(\tau^{\mathrm{adv}}_\theta) = \sum_{h=0}^{H-1} \gamma^h r^{\mathrm{adv}}_h + \gamma^H \mathbb{E}_{a\sim\pi(\cdot|s^{\mathrm{adv}}_H)}\Big[Q^\pi_{\bar\phi}(s^{\mathrm{adv}}_H, a) - \alpha_{\mathrm{ent}} \log \pi(a \mid s^{\mathrm{adv}}_H)\Big]. \tag{21}$$

Rewards after true termination and the terminal bootstrap are zero. The losses used in Line 5 are

$$\mathcal{L}^{\mathrm{rob}}_Q(\phi; \mathcal{B}) = \alpha_{\mathrm{CQL}}\, \mathbb{E}_{\mathcal{B}}\Big[\log \int_{\mathcal{A}} \exp\big(Q^\pi_\phi(s_0, a)\big)\, \mathrm{d}a - Q^\pi_\phi(y_{\mathcal{D}})\Big] + \frac{1}{2}\mathbb{E}_{\mathcal{B}}\Big[\big(Q^\pi_\phi(y_{\mathcal{D}}) - E(\tau^{\mathrm{adv}}_\theta)\big)^2\Big], \tag{22}$$

$$\mathcal{L}_\pi(\vartheta_\pi; \mathcal{B}) = \mathbb{E}_{s_0\sim\mathcal{B},\, a\sim\pi(\cdot|s_0)}\left[\alpha_{\mathrm{ent}} \log \pi(a \mid s_0) - Q^\pi_\phi(s_0, a)\right]. \tag{23}$$

Here $\mathbb{E}_{\mathcal{B}}$ is the empirical average over the supplied batch, and $s_0 \sim \mathcal{B}$ uses its real initial states. The first term retains CQL's conservative regularizer: its action integral becomes a sum for discrete actions and is estimated by importance sampling for continuous actions. The positive coefficients α_{CQL} and α_{ent} control conservatism and entropy, respectively; they are held fixed in this presentation.

GURO−RL replaces only the critic's one-step regression target by Eq. (21). The policy and target critic at the start of the update form the H-step targets. These targets, including any sampled terminal actions, are held fixed during critic regression. The subsequent SAC-style policy update fixes the updated critic's parameters and differentiates through policy-sampled actions at real initial states. Finally, the target critic is softly updated.

Algorithm 3 CQL based GURO−RL: Line 5 update

Require: Adversarial batch $\mathcal{B} = \{(y_{\mathcal{D},i}, \tau^{\mathrm{adv}}_{\theta,i})\}_{i=1}^{B}$ from Line 4 of Algorithm 1; current CQL parameters $\phi, \bar\phi, \vartheta_\pi$; coefficients $\alpha_{\mathrm{CQL}}, \alpha_{\mathrm{ent}} > 0$; learning rates η_Q, η_π; target-update rate ρ

1: Form and fix each target $E(\tau^{\mathrm{adv}}_{\theta,i})$ using Eq. (21)
2: $\phi \leftarrow \phi - \eta_Q \nabla_\phi \mathcal{L}^{\mathrm{rob}}_Q(\phi; \mathcal{B})$
3: $\vartheta_\pi \leftarrow \vartheta_\pi - \eta_\pi \nabla_{\vartheta_\pi} \mathcal{L}_\pi(\vartheta_\pi; \mathcal{B})$, with critic parameters fixed
4: $\bar\phi \leftarrow (1-\rho)\bar\phi + \rho\phi$
5: **return** updated $\phi, \bar\phi, \vartheta_\pi$

B.3 BCQ INSTANTIATIONS OF GURO−RL

Algorithm 4 instantiates Line 5 of Algorithm 1 with BCQ (Fujimoto et al., 2019). Lines 1–4 are unchanged and supply $\mathcal{B} = \{(y_{\mathcal{D},i}, \tau^{\mathrm{adv}}_{\theta,i})\}_{i=1}^{B}$, with $y_{\mathcal{D},i} = (s_{0,i}, a_{0,i})$. BCQ trains its behavior model and perturbation policy using the real initial contexts in this batch, while its critics use the generated continuations to construct adversarial H-step targets.

The conditional VAE $G_{\mathrm{VAE}}(\cdot \mid s)$ models dataset actions given s. Let $\omega = (\omega_1, \omega_2)$ parameterize its encoder $\mathrm{Enc}_{\omega_1}(s, a)$ and decoder $\mathrm{Dec}_{\omega_2}(s, z)$. For each real pair, the encoder outputs $(\mu_z, \sigma_z) = \mathrm{Enc}_{\omega_1}(s_0, a_0)$. Following BCQ's notation, $\mathcal{N}(\mu_z, \sigma_z)$ denotes a diagonal Gaussian with mean μ_z and elementwise standard deviation σ_z; $\mathcal{N}(0, I)$ is the latent prior. The VAE loss is

$$\mathcal{L}_{\mathrm{VAE}}(\omega; \mathcal{B}) = \mathbb{E}_{\mathcal{B}}\Big[\mathbb{E}_{z^{\mathrm{rec}}\sim\mathcal{N}(\mu_z,\sigma_z)} \|a_0 - \mathrm{Dec}_{\omega_2}(s_0, z^{\mathrm{rec}})\|_2^2 + \frac{1}{2d_z} D_{\mathrm{KL}}(\mathcal{N}(\mu_z, \sigma_z) \| \mathcal{N}(0, I))\Big], \tag{24}$$

where $d_z = \dim(z) = 2\dim(\mathcal{A})$ is the latent dimension. The decoder samples $a = \mathrm{Dec}_{\omega_2}(s, z)$ for $z \sim \mathcal{N}(0, I)$.

BCQ adjusts each decoded action with a bounded residual: $g_{\mathrm{pert}}(s, a) = a + \Delta_{\vartheta_{\mathrm{pert}}}(s, a, \Phi_{\mathrm{BCQ}})$, whose residual components lie in $[-\Phi_{\mathrm{BCQ}}, \Phi_{\mathrm{BCQ}}]$. The perturbation bound Φ_{BCQ} is distinct from the flow map Φ_k in the main text. The perturbation parameters and their target copy are $\vartheta_{\mathrm{pert}}$ and $\bar{\vartheta}_{\mathrm{pert}}$. The two critics $Q^{\pi}_{\phi_j}$, $j = 1, 2$, have target copies $Q^{\pi}_{\bar{\phi}_j}$, with $\phi = (\phi_1, \phi_2)$ and $\bar{\phi} = (\bar{\phi}_1, \bar{\phi}_2)$.

After the VAE update, sample and cache $\mathcal{Z} = \{z^{(m)} \sim \mathcal{N}(0, I)\}_{m=1}^{N_{\mathrm{cand}}}$ for each supplied trajectory, suppressing its sample index. The target perturbation model $\bar{g}_{\mathrm{pert}}$ defines the candidate set

$$\mathcal{C}_{\mathrm{BCQ}}(s) = \left\{ \bar{g}_{\mathrm{pert}}\Big(s, \mathrm{Dec}_{\omega_2}(s, z^{(m)})\Big) \right\}_{m=1}^{N_{\mathrm{cand}}}. \tag{25}$$

Using the two target critics, each generated continuation in $\mathcal{B}$ has the adversarial H-step target

$$\begin{aligned} E(\tau_\theta^{\mathrm{adv}}) = \sum_{h=0}^{H-1} \gamma^h r_h^{\mathrm{adv}} + \gamma^H \max_{a \in \mathcal{C}_{\mathrm{BCQ}}(s_H^{\mathrm{adv}})} \Big[\lambda_{\mathrm{BCQ}} \min_{j=1,2} Q^{\pi}_{\bar{\phi}_j}(s_H^{\mathrm{adv}}, a) \\ + (1 - \lambda_{\mathrm{BCQ}}) \max_{j=1,2} Q^{\pi}_{\bar{\phi}_j}(s_H^{\mathrm{adv}}, a) \Big]. \end{aligned} \tag{26}$$

The terminal bootstrap retains BCQ's candidate-action selection and weighted clipped-double-Q backup from its Eq. (13) (Eq. (27) in the supplementary material), with mixture coefficient $\lambda_{\mathrm{BCQ}} \in [0, 1]$. Rewards after true termination and the terminal bootstrap are zero. The critic and perturbation losses used in Line 5 are

$$\mathcal{L}_Q^{\mathrm{rob}}(\phi; \mathcal{B}) = \mathbb{E}_{\mathcal{B}, \mathcal{Z}} \left[\sum_{j=1}^{2} \Big(E(\tau_\theta^{\mathrm{adv}}) - Q^{\pi}_{\phi_j}(y_{\mathcal{D}}) \Big)^2 \right], \tag{27}$$

$$\mathcal{L}_{\mathrm{pert}}(\vartheta_{\mathrm{pert}}; \mathcal{B}) = -\mathbb{E}_{s_0 \sim \mathcal{B},\, a \sim G_{\mathrm{VAE}}(\cdot | s_0)} \left[Q^{\pi}_{\phi_1}(s_0, g_{\mathrm{pert}}(s_0, a)) \right]. \tag{28}$$

Here $\mathbb{E}_{\mathcal{B}, \mathcal{Z}}$ averages over the supplied batch and candidate latent draws, using one cached set per sample. The VAE and perturbation losses use only the real initial contexts in $\mathcal{B}$. Generated continuations, cached candidates, and their H-step targets are fixed during critic regression. The perturbation update fixes the VAE and updated critic parameters, while differentiating through the adjusted action. The target critic and target perturbation parameters are then softly updated.

Policy extraction retains BCQ's candidate-based rule: sample $a^{(m)} \sim G_{\mathrm{VAE}}(\cdot \mid s)$ and select

$$\pi(s) = \operatorname*{arg\,max}_{a \in \{g_{\mathrm{pert}}(s, a^{(m)})\}_{m=1}^{N_{\mathrm{cand}}}} Q^{\pi}_{\phi_1}(s, a). \tag{29}$$

Thus, `GURO-RL` changes the critic regression target while preserving the VAE objective, bounded action perturbation, and policy-extraction rule.

Algorithm 4 `BCQ` based `GURO-RL`: Line 5 update

Require: Adversarial batch $\mathcal{B} = \{(y_{\mathcal{D},i}, \tau_{\theta,i}^{\mathrm{adv}})\}_{i=1}^{B}$ from Line 4 of Algorithm 1; current BCQ parameters $\omega, \vartheta_{\mathrm{pert}}, \bar{\vartheta}_{\mathrm{pert}}, \phi, \bar{\phi}$; bound Φ_{BCQ}; candidate count N_{cand}; mixture λ_{BCQ}; target-update rate ρ; learning rates $\eta_{\mathrm{VAE}}, \eta_Q, \eta_{\mathrm{pert}}$

1: $\omega \leftarrow \omega - \eta_{\mathrm{VAE}} \nabla_\omega \mathcal{L}_{\mathrm{VAE}}(\omega; \mathcal{B})$, using real pairs $(s_{0,i}, a_{0,i})$
2: For each i, sample and cache $\mathcal{Z}_i = \{z_i^{(m)} \sim \mathcal{N}(0, I)\}_{m=1}^{N_{\mathrm{cand}}}$
3: Form candidate sets $\mathcal{C}_{\mathrm{BCQ}}(s_{H,i}^{\mathrm{adv}})$ using Eq. (25)
4: Form and fix each target $E(\tau_{\theta,i}^{\mathrm{adv}})$ using Eq. (26)
5: $\phi \leftarrow \phi - \eta_Q \nabla_\phi \mathcal{L}_Q^{\mathrm{rob}}(\phi; \mathcal{B})$
6: $\vartheta_{\mathrm{pert}} \leftarrow \vartheta_{\mathrm{pert}} - \eta_{\mathrm{pert}} \nabla_{\vartheta_{\mathrm{pert}}} \mathcal{L}_{\mathrm{pert}}(\vartheta_{\mathrm{pert}}; \mathcal{B})$, with VAE and critic parameters fixed
7: $\bar{\phi} \leftarrow (1 - \rho)\bar{\phi} + \rho\phi$
8: $\bar{\vartheta}_{\mathrm{pert}} \leftarrow (1 - \rho)\bar{\vartheta}_{\mathrm{pert}} + \rho\vartheta_{\mathrm{pert}}$
9: **return** updated $\omega, \vartheta_{\mathrm{pert}}, \bar{\vartheta}_{\mathrm{pert}}, \phi, \bar{\phi}$; policy π is defined by Eq. (29)

C Details of MuJoCo Experiment

C.1 Datasets

We employ offline **MuJoCo-V5** datasets from **Minari** for *Hopper* (**H**), *HalfCheetah* (**HC**), and *Walker2d* (**W**) (Younis et al., 2024). For each task, we construct *Simple-Medium* (**SM**) and *Medium-Expert* (**ME**) mixtures by combining equal fractions of the simple and medium datasets, and of the medium and expert datasets, respectively. Mixing is performed at the trajectory level, preserving temporal consistency for multi-step targets. Task–dataset settings are abbreviated as, e.g., **H-SM**. Table 3 summarizes the sizes of all offline dataset mixtures used in our experiments.

Task	Episodes	Transitions
H-SM	1,639	992,464
H-ME	1,206	1,002,056
W-SM	1,030	1,002,965
W-ME	1,022	1,003,137
HC-SM	1,000	1,000,000
HC-ME	1,000	1,000,000

Table 3: Sizes of the offline mixing datasets.

C.2 OOD Settings and Shift Measurement

OOD Setting. We adopt **Robust Gymnasium** (Gu et al., 2025) to construct environments with OOD dynamics for the three continuous-control locomotion **MuJoCo-V5** tasks and evaluate the robustness of each baseline. Specifically, we consider three types of dynamics shifts: **Morphology**, **Friction**, and **Gravity**. As shown in Table 4, we modify one parameter family at a time by multiplying its nominal values by a scaling factor, while keeping all other settings fixed. Each shift family uses four scaling factors, $(0.8, 0.9, 1.1, 1.2)$, with a factor of 1 corresponding to the nominal environment.

Table 4: Nominal and OOD settings for the MuJoCo experiments.

Scenario	Task	Type	Assigned Values
Nominal	All tasks	All dynamic parameters	Default
Morphology	Hopper	Torso/foot lengths ($\mathbf{m}_{\mathrm{TF}}{}^{*}$)	$\mathbf{m}_{\mathrm{TF}} \times (0.8, 0.9, 1.1, 1.2)$
	Walker2d	Torso/foot lengths ($\mathbf{m}_{\mathrm{TF}}{}^{*}$)	$\mathbf{m}_{\mathrm{TF}} \times (0.8, 0.9, 1.1, 1.2)$
	HalfCheetah	Back/front foot lengths ($\mathbf{m}_{\mathrm{BF}}{}^{*}$)	$\mathbf{m}_{\mathrm{BF}} \times (0.8, 0.9, 1.1, 1.2)$
Friction	All tasks	Contact-friction coefficients ($\mathbf{f}^{*}$)	$\mathbf{f} \times (0.8, 0.9, 1.1, 1.2)$
Gravity	All tasks	Gravity vector ($\mathbf{g}^{*}$)	$\mathbf{g} \times (0.8, 0.9, 1.1, 1.2)$

* $\mathbf{m}_{\mathrm{TF}}$, $\mathbf{m}_{\mathrm{BF}}$, $\mathbf{f}$, and $\mathbf{g}$ denote task-specific default parameter values.

OOD Shift Measurement. The OOD settings involve different types of dynamics parameter changes, making it difficult to quantify the resulting shifts directly. We therefore compare nominal and OOD state distributions to characterize these shifts and inform the choice of the ambiguity budget for each task.

Specifically, for each task and offline dataset mixture (**SM** or **ME**), we trained all the baselines on that mixture with different random seeds and recorded their nominal trajectories. We then replay the recorded actions in each OOD environment from the same initial states, ensuring $s_0^{\mathrm{OOD}} = s_0^{\mathrm{nom}}$ and $a_t^{\mathrm{OOD}} = a_t^{\mathrm{nom}}$, while subsequent states may diverge. We collect 20,000 state observations per environment and compare the empirical state distributions using the W_2 distance, with squared Euclidean transport cost in the original state space.

Task	$\bar{D}$	$\bar{D}_{\mathrm{ME}}/\bar{D}_{\mathrm{SM}}$
H-SM	0.057	2.0×
H-ME	0.115	
W-SM	0.107	1.7×
W-ME	0.178	
HC-SM	1.578	1.3×
HC-ME	2.029	

Table 5: Average W_2 distances between the nominal setting and all OOD dynamics settings for each task.

Let $\mathcal{I}$ index all combinations of evaluated methods, training seeds, and OOD settings. For $\iota \in \mathcal{I}$, let P_ι^{OOD} and P_ι^{nom} denote the paired empirical state distributions. The mean shift is defined as

$$\bar{D} = \frac{1}{|\mathcal{I}|} \sum_{\iota \in \mathcal{I}} W_2\big(P_\iota^{\mathrm{OOD}}, P_\iota^{\mathrm{nom}}\big)\,. \tag{30}$$

Table 5 reports $\bar{D}$. For a given task, **SM** and **ME** share the same OOD perturbations, but their learned policies may generate different action sequences and therefore yield different measured state-distribution shifts. Within every robot the **ME** regime experiences a larger shift than **SM**—$2.0\times$ on Hopper, $1.7\times$ on Walker2d, and $1.3\times$ on HalfCheetah. The reason is that a stronger policy operates in a narrower, faster region of transition space, where the same simulator perturbation displaces the visited transitions further.

C.3 Baselines

We compare `GURO-RL` with the following baselines. For a controlled and fair comparison, we adapt the core mechanisms of compatible baselines to a shared backbone.

`IQL`: `IQL` (Kostrikov et al., 2021) learns offline policies through expectile value regression and advantage-weighted behavior cloning. It serves as the model-free reference and provides the shared backbone for `ATLA`, `RSC`, `ROMB`, `FWM` and `PLR-PVL`.

`FWM`: `FWM` is our flow-matching adaptation of `DWM` Ding et al. (2024), replacing its diffusion model with the same pretrained conditional flow model used by `GURO-RL` to generate non-adversarial multi-step rollouts for value learning.

`ATLA`: `ATLA` Zhang et al. (2021) alternates policy and observation-adversary training. Our offline `IQL` adaptation trains the adversary on offline states with a regularized critic-based objective. With adversarial loss weight $\lambda_{\text{adv}} \in [0, 1]$ and ℓ_∞ perturbation budget $\varepsilon_{\text{ATLA}} > 0$, ATLA actor updates minimize

$$\mathcal{L}_\pi^{\text{ATLA}}(\pi) = -\mathbb{E}_{(s,a)\sim\mathcal{D}}[w_{\text{IQL}}(s,a)\left((1-\lambda_{\text{adv}})\log\pi(a \mid s) + \lambda_{\text{adv}}\log\pi(a \mid s+\delta(s))\right)]. \quad (31)$$

Here, w_{IQL} is the clipped clean-sample advantage weight, and $\delta(s)$ is the adversary's deterministically squashed and scaled Gaussian mean, satisfying $\|\delta(s)\|_\infty \le \varepsilon_{\text{ATLA}}$. Value and critic updates use clean offline data.

`RFQI`: `RFQI` (Panaganti et al., 2022) combines behavior-constrained action selection with a total-variation-robust Bellman backup. We train it from scratch on the same offline dataset, independently of `IQL`.

`RSC`: `RSC` (Ding et al., 2023) is an online, off-policy method that learns a sparse causal dynamics mask from replay data, rewrites low-importance state coordinates, and regenerates next states and rewards with a learned causal transition model. The synthetic transitions are mixed into the buffer for SAC updates. With a fixed offline dataset and no new interactions to refit the mask, we estimate it only once and thus `RSC` retains causal identification of spurious coordinates as input augmentation while preserving `IQL` losses.

`ROMB`: `ROMB` Chen et al. (2026) jointly learns a robust policy and an adversarial ensemble of Gaussian one-step dynamics models through KL-constrained Stackelberg optimization. In our `IQL` adaptation, we pretrain the ensemble on the fixed offline dataset, augment critic regression with model-generated transitions, and add the rollout policy gradient with a Stackelberg correction to the actor update, leaving the value update unchanged.

`PLR-PVL`: `PLR-PVL` Berdica et al. (2024) uses positive value loss (PVL) as a regret proxy to prioritize replay across pretrained world models. Our adaptation uses three frozen flow models pretrained on the same offline dataset with different random initializations and data shuffling orders. At each update, we sample a model from a PVL-based distribution, use its generated trajectory to form the `IQL` critic's H-step target, and update the distribution with the trajectory's PVL score. The `IQL` value and actor losses remain unchanged.

C.4 Experimental Setup

C.4.1 Training Setup

Flow Model Setup. Our flow model uses a conditional 2D U-Net with encoder widths of 128 and 256 and skip connections. Its main convolutional blocks use 3×3 kernels, batch normalization, and GELU activations. Separate two-layer MLPs encode the initial state–action pair and flow time. Let T_{win} denote the trajectory window length. The model also supports an optional return-to-go

condition, $g_t = \sum_{h=0}^{T_{\text{win}}-1} \gamma^h r_{t+h}$, encoded by two-layer MLPs and used only by FWM to guide trajectory generation.

During pretraining, we extract complete within-episode windows for each task and offline dataset mixture (**SM/ME**), using $T_{\text{win}} = 12$ for Hopper and $T_{\text{win}} = 18$ for Walker2d and HalfCheetah. We normalize states using empirical marginal CDFs and min–max scale rewards and, for FWM, g_t labels to $[-1, 1]$. We pretrain the model for $5{,}000$ training epochs using Adam with a learning rate of 10^{-4} and a batch size of 256. Conditioning inputs are dropped with probability 0.2. Generation uses 5 midpoint ODE steps with a classifier-free guidance weight of 2.

Baselines Setup. Backbone symbols follow Appendix B; the settings below use the same learning-rate and coefficient notation.

IQL: For IQL and the baselines using it as a backbone, we use a Gaussian actor, two independent Q-networks, and a state-value network, each implemented as an MLP with two hidden layers of 256 units and ReLU activations. For IQL, we use a discount factor $\gamma = 0.99$, batch size $B = 256$, and Adam learning rates $\eta_\pi = \eta_Q = \eta_V = 3 \times 10^{-4}$. The target-network update rate is $\rho = 0.005$, the expectile parameter is $\alpha_{\text{IQL}} = 0.7$, and the advantage-weight inverse temperature is $\beta_{\text{IQL}} = 3$.

GURO-RL:

- **IQL based GURO-RL.** As our default setting, we use an IQL backbone in GURO-RL and fine-tune the pretrained flow model. For **H-SM**, **H-ME**, **W-SM**, **W-ME**, **HC-SM**, and **HC-ME**, respectively, the DRO budgets ϵ are $(0.35, 0.70, 0.05, 0.50, 0.10, 0.50)$, the Lagrange multipliers μ are $(0.5, 0.1, 0.7, 0.03, 0.3, 0.05)$, and the PPO clipping parameters ϵ_{ppo} are $(0.2, 0.4, 0.2, 0.4, 0.4, 0.8)$. The flow fine-tuning learning rate and decay factor are $(10^{-5}, 0.99)$ for **H-SM**, $(5 \times 10^{-5}, 0.995)$ for **H-ME**, **W-SM**, and **W-ME**, and $(10^{-4}, 0.999)$ for **HC-SM** and **HC-ME**. Each inner optimization runs for 2 rounds, except for **HC-ME**, which uses 4. All settings use a dual step size of $\eta_\mu = 5 \times 10^{-3}$, a rollout horizon of $H = 5$, and $\beta_Q = 0.1$.
- **CQL based GURO-RL.** In the backbone ablation, we instantiate GURO-RL with a CQL backbone. For **H-SM**, **W-SM**, and **HC-SM**, respectively, batch sizes B are $(256, 256, 1024)$ and conservative weights α_{CQL} are $(1, 1, 5)$. All settings use a conservative-regularizer temperature of 1, Adam learning rates $\eta_Q = \eta_\pi = 3 \times 10^{-4}$, and a learning rate of 3×10^{-4} for the entropy coefficient α_{ent}; **HC-SM** also uses a max-Q backup over 10 actions. In the same order, we set DRO budgets ϵ to $(0.35, 0.05, 0.01)$, initialize multipliers μ to $(0.5, 0.7, 0.3)$, and use PPO clipping parameters ϵ_{ppo} of $(0.2, 0.2, 0.4)$, flow fine-tuning learning rates of $(10^{-5}, 5 \times 10^{-5}, 10^{-4})$, and decay factors of $(0.99, 0.995, 0.999)$. The dual step size is $\eta_\mu = 5 \times 10^{-3}$ throughout.
- **BCQ based GURO-RL** In the backbone ablation, we also instantiate GURO-RL with a BCQ backbone. All settings use normalized states, batch size $B = 256$, and Adam learning rates $\eta_{\text{VAE}} = \eta_Q = \eta_{\text{pert}} = 3 \times 10^{-4}$. The behavior VAE G_{VAE} has a hidden width of 750 and latent dimension $d_z = 2\dim(\mathcal{A})$. We set the action perturbation limit $\Phi_{\text{BCQ}} = 0.05$ and clipped-double-Q mixture weight $\lambda_{\text{BCQ}} = 0.75$, using $N_{\text{cand}} = 10$ during training and $N_{\text{cand}} = 100$ during evaluation. For **H-SM**, **W-SM**, and **HC-SM**, respectively, DRO budgets ϵ are $(0.35, 0.05, 1.0)$, multipliers μ are $(0.5, 0.7, 0.3)$, and dual step sizes η_μ are $(5 \times 10^{-3}, 5 \times 10^{-3}, 5 \times 10^{-2})$. Flow fine-tuning uses learning rates of $(10^{-5}, 5 \times 10^{-5}, 10^{-3})$, decay factors of $(0.99, 0.995, 0.999)$, and PPO clipping parameters ϵ_{ppo} of $(0.2, 0.2, 0.4)$.

FWM: For FWM, we use IQL as the backbone with hyperparameter settings, while keeping the pretrained flow model frozen. We set the rollout horizon to $H = 5$ and compute g_t from each real T_{win}-step trajectory in the minibatch. Together with the state-action pair, g_t conditions the generation of state-reward trajectories used to construct H-step targets for the IQL critic.

ATLA: For ATLA, we use IQL as the backbone with its hyperparameter settings. We set $\varepsilon_{\text{ATLA}} = 0.025$ and $\lambda_{\text{adv}} = 0.1$, and train the adversary that generates $\delta(s)$ with a learning rate of 3×10^{-4}.

RFQI: We train RFQI from scratch using its native procedure, without an IQL backbone or initialization. The ambiguity radius, actor and critic learning rates are 0.5, 3×10^{-4}, and 6×10^{-4}, respectively (0.6, 10^{-4}, and 8×10^{-4} for W-ME). All settings use a batch size of $1{,}000$, with dual-

network and behavior-VAE learning rates of 3×10^{-4} and 10^{-3}. The action perturbation limit is 0.05, and the clipped-double-Q mixture weight is 0.75. Per state, the VAE generates 10 candidate actions during training and 30 during evaluation.

RSC: For `RSC`, we use an `IQL` backbone with hyperparameter settings and pretrain a causal model (four 256-unit hidden layers, Gumbel–Bernoulli coordinate mask) on each offline mixture for 50 epochs at a learning rate of 10^{-4}, then freeze it. The model predicts next states and rewards and learns a sparse mask to select low-importance state coordinates. Training applies causal permutation within minibatches with a rewrite ratio of 0.75 and importance threshold of 0.50, followed by unchanged `IQL` updates on the rewritten batch.

ROMB: For `ROMB`, we use an `IQL` backbone with its hyperparameter settings and generate rollouts with 5 Gaussian dynamics models, each with four hidden layers of 200 units, pretrained on the offline datasets for 30 epochs at a learning rate of 10^{-3}. We set the rollout horizon to $H = 5$, the rollout batch size to 256, the Generalized Advantage Estimation (GAE) coefficient to 0.95, the PPO clip range to 0.2 and the Stackelberg correction coefficient to 10^{-8}. The second-order horizon cap and model constraint threshold are both 10.

PLR−PVL: For `PLR-PVL`, we use an `IQL` backbone with its hyperparameter settings and an ensemble of 3 pretrained, frozen flow models. We score their generated trajectories over $H = 5$ steps using GAE-based positive value loss with a GAE coefficient of 0.95.

Training Protocol. We first pretrain the flow models on each fixed offline dataset as described above. For each task and training seed (0,42,3047), Stage-1 pretrains the offline RL backbone, providing a shared initialization for compatible methods. To improve training efficiency, Stage-2 continues training on the same dataset with one method-specific update every 10 training iterations and standard backbone updates otherwise. Early stopping is applied during Stage 2. Table 6 lists the training epochs for both stages.

Task	Stage-1 Epochs	Stage-2 Epochs
H-SM	200	50
H-ME	1,500	375
W-SM	200	50
W-ME	400	100
HC-SM	3,000	750
HC-ME	10,000	2,500

Table 6: Training epochs for each task.

Evaluation Protocol. We measure performance using the normalized score defined as $\mathrm{Score}(\bar{R}) = 100\,(\bar{R} - R_{\min})/(R_{\max} - R_{\min})$. Here, $\bar{R}$ is the mean episodic return, $R_{\min}$ is the D4RL random-policy reference return (Fu et al., 2020), and $R_{\max}$ is the mean undiscounted episodic return in the corresponding Minari `expert-v0` dataset (Younis et al., 2024). Table 7 lists the reference values.

Task	$R_{\min}$	$R_{\max}$
Hopper	-20.272	3,857.80
Walker2d	1.629	6,847.79
HalfCheetah	-280.179	16,242.92

Table 7: Reference returns for score normalization.

For each of three independent training seeds, we evaluate the final five checkpoints for 10 episodes per environment, each capped at $1{,}000$ steps. Within each seed, we average normalized scores over checkpoints and episodes for each environment, then across environments within each OOD family. We report the mean and sample standard deviation of these seed-level scores across the three seeds.

D Additional results

D.1 Main results

Our MuJoCo experiments compare the robustness of `GURO-RL` and the baselines under nominal dynamics and several types of OOD dynamics shifts across both **SM** and **ME** datasets. As shown in Table 8, `GURO-RL` achieves the highest all-scenario average in all six task–dataset settings across **SM** and **ME**. Its overall and OOD averages reach 64.5 and 57.1, exceeding `IQL` by 9.5 and 9.1 points and the strongest baseline averages by 7.2 and 6.7 points, respectively. These gains accompany higher nominal scores than `IQL` in all six settings, including the highest nominal scores among

Task	Scenario	Offline RL	World Model	Robust RL			Robust RL with Generative Model		
		IQL	FWM	ATLA	RFQI	RSC	ROMB	PLR-PVL	GURO-RL
H-SM	Nominal	84.3 ±4.8	87.7 ±5.0	82.2 ±7.7	83.6 ±3.4	89.1 ±2.4	87.6 ±5.8	87.2 ±4.4	89.9 ±2.4
	Morphology	49.2 ±0.8	48.4 ±4.6	49.8 ±4.5	52.5 ±1.0	49.8 ±2.7	51.6 ±2.2	48.3 ±8.3	48.1 ±7.9
	Friction	42.7 ±11.3	43.8 ±5.3	52.8 ±4.9	35.0 ±3.7	49.2 ±13.0	47.8 ±7.0	46.3 ±6.6	53.1 ±8.5
	Gravity	60.0 ±17.5	64.7 ±7.7	68.2 ±11.2	45.7 ±3.2	67.2 ±13.4	64.2 ±14.4	65.9 ±13.5	72.0 ±13.0
	Average	59.1	61.2	63.3	54.2	63.8	62.8	61.9	65.8
H-ME	Nominal	92.9 ±9.5	95.9 ±2.4	92.5 ±11.3	63.3 ±23.9	94.1 ±11.2	92.1 ±4.3	93.8 ±6.0	93.3 ±8.5
	Morphology	38.8 ±3.4	37.6 ±5.7	41.0 ±1.6	55.2 ±1.2	41.2 ±1.7	39.2 ±2.4	43.1 ±2.3	46.7 ±5.0
	Friction	22.8 ±2.8	22.9 ±2.9	26.9 ±1.4	33.8 ±3.3	24.6 ±3.3	26.2 ±2.2	25.6 ±2.6	37.8 ±14.3
	Gravity	35.7 ±6.2	38.7 ±3.2	35.9 ±5.7	47.8 ±7.4	37.5 ±5.0	37.6 ±3.3	39.0 ±5.0	45.0 ±8.6
	Average	47.5	48.8	49.1	50.0	49.3	48.8	50.4	55.7
W-SM	Nominal	55.4 ±6.2	68.7 ±9.2	61.6 ±13.6	56.0 ±1.4	71.8 ±2.9	57.8 ±5.5	60.6 ±9.7	75.4 ±5.2
	Morphology	55.7 ±5.1	58.5 ±6.6	56.8 ±1.2	51.6 ±0.4	56.6 ±2.6	59.1 ±4.8	55.5 ±7.6	61.4 ±9.5
	Friction	57.5 ±5.7	64.0 ±6.6	61.6 ±9.8	56.8 ±0.9	67.0 ±4.6	60.9 ±5.7	60.9 ±9.5	65.0 ±8.9
	Gravity	45.0 ±1.2	45.5 ±3.9	44.4 ±0.8	53.1 ±0.8	45.3 ±2.1	44.7 ±2.3	43.6 ±1.8	50.1 ±1.8
	Average	53.4	59.2	56.1	54.4	60.2	55.6	55.1	63.0
W-ME	Nominal	87.4 ±7.6	89.6 ±3.5	85.4 ±9.8	79.7 ±3.5	89.2 ±4.1	95.1 ±4.6	87.2 ±8.3	92.3 ±3.1
	Morphology	54.4 ±5.4	53.7 ±4.9	56.6 ±4.5	51.9 ±2.4	53.7 ±0.9	54.4 ±4.1	55.3 ±8.5	67.3 ±1.8
	Friction	68.4 ±4.2	69.1 ±3.3	69.6 ±6.3	76.3 ±1.3	68.4 ±3.7	72.9 ±3.5	71.3 ±4.1	77.7 ±4.0
	Gravity	27.7 ±3.2	27.4 ±3.9	31.9 ±4.8	52.4 ±2.5	29.7 ±2.0	29.6 ±2.2	27.3 ±1.9	38.0 ±4.8
	Average	59.5	59.9	60.9	65.1	60.3	63.0	60.3	68.8
HC-SM	Nominal	62.7 ±4.2	61.8 ±0.7	64.9 ±1.6	47.6 ±3.4	65.3 ±6.0	65.0 ±1.3	64.3 ±2.6	74.4 ±3.8
	Morphology	54.5 ±1.0	54.8 ±1.9	53.8 ±2.6	44.4 ±1.4	55.0 ±3.5	54.3 ±1.3	55.8 ±2.8	54.2 ±0.3
	Friction	63.7 ±2.4	65.1 ±2.4	61.0 ±0.3	46.0 ±2.1	64.0 ±4.1	64.3 ±2.4	63.7 ±5.3	70.2 ±4.3
	Gravity	43.9 ±2.6	44.6 ±2.8	43.4 ±2.0	43.0 ±0.7	46.7 ±3.4	44.8 ±3.6	45.7 ±1.7	47.7 ±1.3
	Average	56.2	56.6	55.8	45.2	57.8	57.1	57.4	61.6
HC-ME	Nominal	72.6 ±5.0	71.0 ±10.7	79.2 ±7.3	74.3 ±3.0	67.7 ±5.7	77.6 ±5.3	85.0 ±6.9	95.6 ±0.7
	Morphology	37.6 ±5.1	38.1 ±6.4	36.5 ±2.0	45.9 ±3.5	37.4 ±2.0	39.4 ±2.0	37.6 ±2.2	52.0 ±7.2
	Friction	73.8 ±5.3	70.3 ±10.9	75.6 ±10.6	70.8 ±2.8	69.9 ±2.9	76.5 ±6.0	74.7 ±4.7	93.1 ±2.0
	Gravity	33.5 ±1.2	31.0 ±4.1	32.0 ±5.1	45.7 ±4.8	33.7 ±3.9	31.9 ±1.7	32.3 ±2.7	48.0 ±0.6
	Average	54.4	52.6	55.9	59.2	52.2	56.3	57.4	72.2
Average	OOD	48.0	48.8	49.9	50.4	49.8	50.0	49.5	57.1
	ALL	55.0	56.4	56.8	54.7	57.3	57.3	57.1	64.5

Table 8: Full MuJoCo main comparison performance.

all methods in the three **SM** tasks. The OOD improvement over IQL is larger in **ME**: while IQL's OOD average decreases from 52.5 in **SM** to 43.6 in **ME**, GURO-RL maintains averages of 58.0 and 56.2, respectively. This comparison suggests that expert trajectories alone do not ensure robustness to dynamics shifts and that explicit uncertainty modeling remains beneficial across dataset mixtures.

Across all six task–dataset settings, GURO-RL exceeds FWM by 8.3 points in mean OOD score, with the gap increasing from 3.6 in **SM** to approximately 13.0 in **ME**. This pattern suggests that frozen multi-step generation alone does not necessarily translate stronger offline demonstrations into robustness against dynamics shifts. By adapting the generator against the current policy, GURO-RL searches for adverse continuations that challenge its value estimates; IPW directs this search toward policy-relevant observed actions, while the flow-matching constraint limits departure from the offline distribution. Compared with ROMB and PLR-PVL, GURO-RL achieves OOD improvements of 7.1 and 7.6 points, respectively. The gain over ROMB may reflect the greater flexibility of our flow-based uncertainty set in representing high-dimensional state–reward distributions compared with its explicit Gaussian transition models. Unlike PLR-PVL's reweighting of a fixed pretrained ensemble, adversarial fine-tuning updates the generator itself, allowing exploration beyond the variations already represented by the frozen candidates. It also outperforms RSC and ATLA by 7.3 and 7.2 OOD points, with the advantage over RSC widening from 2.4 in **SM** to approximately 12.2 in **ME**. The additional gains may arise because GURO-RL trains on generated multi-step trajectories designed to expose weaknesses of the current policy, whereas RSC and ATLA mainly perturb input states. The larger gaps in **ME** are consistent with the benefit of directly addressing dynamics uncertainty rather than relying on input augmentation alone. Relative to RFQI, GURO-RL improves the mean OOD score by 6.7 points. Independent worst-case backups may be overly pessimistic under

Task	Scenario	IQL Backbone		CQL Backbone		BCQ Backbone	
		IQL	GURO-RL	CQL	GURO-RL	BCQ	GURO-RL
H-SM	Nominal	84.3 ±4.8	89.9 ±2.4	91.7 ±0.3	91.7 ±0.3	84.6 ±2.8	86.8 ±6.0
	Morphology	49.2 ±0.8	48.1 ±7.9	54.8 ±0.3	55.1 ±0.8	29.0 ±7.7	32.1 ±1.0
	Friction	42.7 ±11.3	53.1 ±8.5	74.7 ±1.1	75.4 ±0.6	42.2 ±5.8	47.5 ±2.4
	Gravity	60.0 ±17.5	72.0 ±13.0	66.3 ±5.7	67.6 ±8.3	56.0 ±2.3	60.3 ±1.1
	Average	59.1	65.8	71.9	72.4	52.9	56.7
W-SM	Nominal	55.4 ±6.2	75.4 ±5.2	33.6 ±16.0	44.5 ±11.8	65.3 ±5.0	66.0 ±2.3
	Morphology	55.7 ±5.1	61.4 ±9.5	18.3 ±4.9	20.3 ±2.6	56.7 ±3.7	59.9 ±3.6
	Friction	57.5 ±5.7	65.0 ±8.9	33.5 ±10.5	40.4 ±11.6	62.3 ±3.1	62.6 ±2.4
	Gravity	45.0 ±1.2	50.1 ±1.8	32.6 ±6.0	36.9 ±13.2	56.5 ±3.4	60.2 ±2.7
	Average	53.4	63.0	29.5	35.5	60.2	62.2
HC-SM	Nominal	62.7 ±4.2	74.4 ±3.8	64.0 ±11.5	53.6 ±10.0	51.9 ±3.4	58.1 ±9.5
	Morphology	54.5 ±1.0	54.2 ±0.3	36.5 ±2.3	40.9 ±0.2	49.4 ±1.9	49.4 ±2.8
	Friction	63.7 ±2.4	70.2 ±4.3	56.9 ±0.9	48.9 ±1.7	52.8 ±4.4	54.4 ±4.7
	Gravity	43.9 ±2.6	47.7 ±1.3	33.8 ±1.3	40.8 ±0.6	43.3 ±2.2	46.5 ±1.6
	Average	56.2	61.6	47.8	46.1	49.3	52.1
Average	OOD	52.5	58.0	45.3	47.4	49.8	52.5
	ALL	56.2	63.5	49.7	51.3	54.2	57.0

Table 9: Backbone generality of GURO-RL on Simple-Medium tasks.

coupled physical changes, whereas shared flow parameters restrict arbitrary combinations of local perturbations. Nevertheless, RFQI leads under **H-ME** morphology and gravity shifts and **W-ME** gravity shifts, showing that its robustness mechanism remains effective in particular settings. Thus, the aggregate gains support the usefulness of policy-dependent generative uncertainty, but do not establish uniform superiority or isolate any single mechanism as their cause.

D.2 Ablation Study

Backbone Ablation. Table 9 demonstrates that GURO-RL can improve robustness across backbones with different value estimation and policy learning mechanisms. It increases the OOD average from 52.5 to 58.0 for IQL, from 45.3 to 47.4 for CQL, and from 49.8 to 52.5 for BCQ, while improving the all-scenario average in most of the nine task–backbone combinations. This generality follows from how generative uncertainty enters the learning process: adversarial multi-step trajectories supply critic regression targets, while each backbone retains its native mechanisms for controlling offline extrapolation. Specifically, IQL retains expectile value regression and advantage-weighted behavior cloning, CQL retains its conservative Q-regularizer, and BCQ retains its behavior-constrained action selection. The terminal value estimator connects each backbone to the adversarial search, allowing the same generative uncertainty framework to account for their different value estimates. Thus, robustness to dynamics shifts can complement distinct approaches to handling limited data coverage. The benefits nevertheless depend on the backbone and task: on **HC-SM**, the CQL-based all-scenario average decreases from 47.8 to 46.1, despite improvements under morphology and gravity shifts. One possible explanation is that adverse value targets compound CQL's existing pessimism, suggesting that the uncertainty budget should account for the backbone's degree of conservatism.

Inverse Propensity Weighting Ablation. Figure 1 shows that IPW improves the average score in five of six settings, with the largest gains on **W-SM** and **HC-SM** (3.8 and 3.6 points). Without IPW, the adversarial objective weights initial actions according to their frequency in the offline dataset, which may differ from their relevance to the current policy. Using $w_\pi = \pi(a \mid s)/\pi_0(a \mid s)$, with π_0 approximating the behavior policy, approximately corrects this mismatch at observed states.

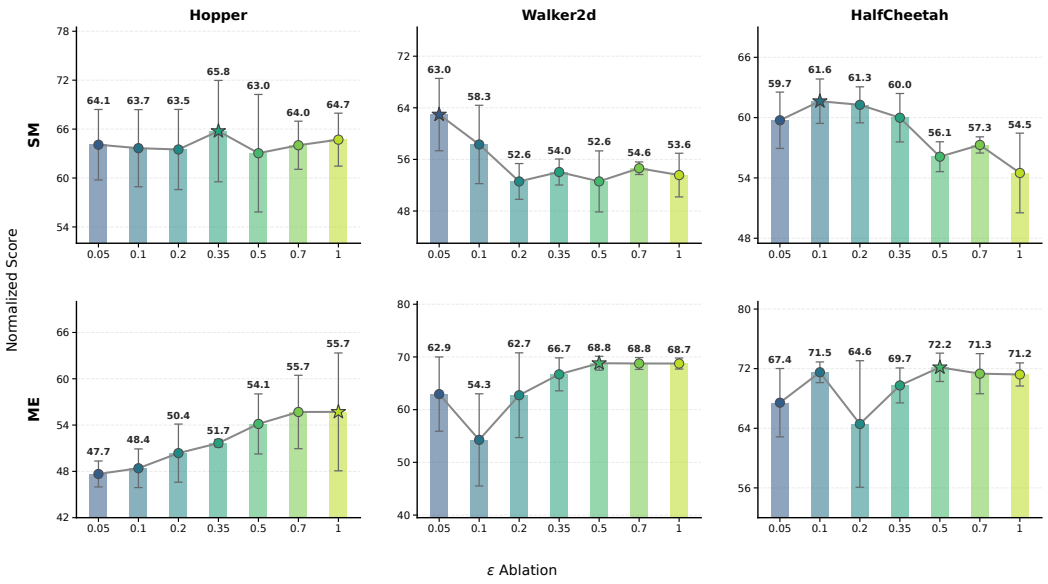

Figure 2: Results ambiguity budget ϵ ablation.

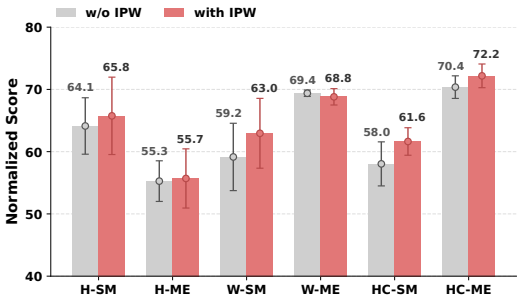

Figure 1: Result of IPW ablation.

Actions favored by the current policy relative to the behavior policy receive greater weight, directing adversarial updates toward dynamics variations that threaten the policy's own decisions. This prioritization matters because the flow model shares parameters across contexts: its updates must balance competing directions of value degradation under a common consistency constraint. IPW therefore helps allocate adversarial search toward relevant vulnerabilities, producing critic targets that better expose optimistic estimates on actions the policy is likely to select, while retaining generation from observed state–action pairs. The larger gains in **SM** are consistent with a greater need to distinguish policy-relevant actions within heterogeneous behavior data. The smaller gains in **ME**, including a 0.6-point decrease on **W-ME**, suggest that the benefit depends on the policy–data mismatch and the accuracy of the estimated weights.

Ambiguity Budget Ablation We examine the sensitivity of GURO−RL to the ambiguity budget ϵ, which controls the allowable deviation of the adversarial model from the offline data distribution. Figure 2 reveals that the preferred budget varies across tasks and datasets. The highest mean scores are obtained at $\epsilon = 0.35$ for **H-SM**, 0.05 for **W-SM**, and 0.1 for **HC-SM**. For **ME**, the best budget is 1.0 for Hopper, 0.7 for Walker2d, and 0.5 for HalfCheetah, with ties at the precision reported in the figure. A budget that is too small restricts adversarial adaptation, potentially leaving harmful dynamics variations unexplored. Conversely, an overly large budget permits substantial departure from the offline distribution and may produce excessively pessimistic value targets. The budget therefore balances the scope of adversarial search against consistency with the data, with the preferred value depending on the task and dataset.

Rollout Horizon Ablation. We examine the effect of the rollout horizon H, which determines how many generated steps contribute to adversarial value estimation. Figure 3 shows that performance depends on the horizon, with moderate values generally providing a favorable balance between multi-step information and model reliability. $H = 5$ achieves the highest mean score on most tasks. Increasing H allows adversarial evaluation to capture longer-term consequences of dynamics

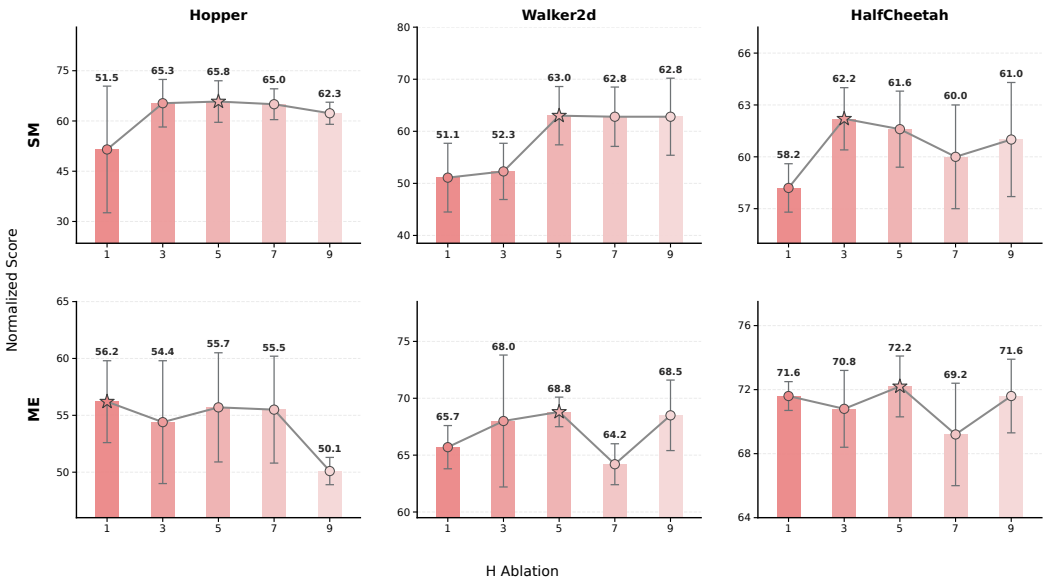

Figure 3: Result of rollout horizon ablation.

variations and reduces the weight γ^H on the terminal bootstrap estimate. However, longer horizons also increase reliance on generated rewards and distant states, making value targets more sensitive to model inaccuracies. The horizon therefore balances multi-step information against model reliability, with $H = 5$ providing a practical default across the evaluated settings.

E CASE STUDY: ROBUST CONTROL OF AN EPANET WATER DISTRIBUTION NETWORK

E.1 ENVIRONMENT

We employ a Water Distribution Network (WDN) as an infrastructure-level robustness evaluation, where policy failures can reduce service quality, violate pressure requirements, or increase pump-energy cost rather than merely lower a benchmark score. The controller must maintain pressure and demand satisfaction under uncertain hydraulic conditions while limiting energy use. We implement `EPAGym`, a Gym-style interface to the EPANET hydraulic simulator through EPYT. Figure 4 shows the network topology.

Dynamics. The network follows the standard EPANET tutorial topology. It contains six demand junctions J2–J7, one reservoir R1, one tank T8, five ordinary pipes $\mathrm{P2}, \mathrm{P3}, \mathrm{P4}, \mathrm{P6}, \mathrm{P8}$, one head-curve pump PUMP, and three pressure-breaking-valve (PBV) links P1, P5, and P7. Junction elevations are 700 ft except for $\mathrm{J3} = 710$ ft and $\mathrm{J5} = 650$ ft; R1 has head 700 ft, and T8 has elevation 830 ft with initial, minimum, and maximum levels $4/0/20$ ft. The nominal demands at J2–J7 are $(0, 150, 150, 200, 150, 120)$ GPM and are modulated by hourly demand patterns. Ordinary pipes use Hazen–Williams roughness 100, lengths 5000–7000 ft, and diameters 6–12 inches. The PBVs have diameters $(12, 12, 6)$ inches and initial pressure-drop settings $(50, 80, 100)$ PSI, while the pump uses head curve C1 with point $(600, 350)$. The control environment uses pressure-driven analysis with a 60-PSI required pressure. At each control step, the agent sets the pump and three PBV actuators; EPANET then solves the coupled hydraulic state under the current nodal demands. The resulting pressures, delivered demands, and pump energy determine the next state and raw environment reward.

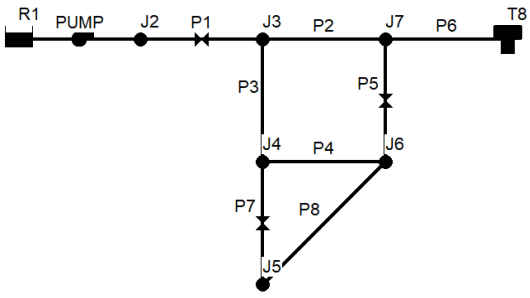

Figure 4: Topology of the water distribution network used in the case study.

Action Space. The WDN action space is

$$a_t^{\mathrm{WDN}} = \left(a_t^{\mathrm{PUMP}}, a_t^{\mathrm{P_1}}, a_t^{\mathrm{P_5}}, a_t^{\mathrm{P_7}}\right) \in \mathcal{A}_{\mathrm{WDN}} = [-1, 1]^4. \tag{32}$$

Each action dimension controls one hydraulic actuator: the pump action controls pump speed or effective capacity, and each PBV action controls its imposed pressure drop.

State Space. Let $\mathcal{V}_{\mathrm{WDN}}$ be the monitored-node set. At environment step t, let $p_{j,t}^{\mathrm{hyd}}$ be the pressure at node j and let $d_{j,t-1}^{\mathrm{dem}}$ be its demand in the preceding step. We use t for the environment clock and reserve k for FM continuous time. The policy state is

$$s_t^{\mathrm{WDN}} = \left[\left(p_{j,t}^{\mathrm{hyd}}, d_{j,t-1}^{\mathrm{dem}}\right)_{j \in \mathcal{V}_{\mathrm{WDN}}}, a_{t-1}^{\mathrm{WDN}}\right]. \tag{33}$$

Reward. The reward balances pump energy, pressure regulation, demand satisfaction, and action smoothness. Let e_t^{pump} denote mean pump energy and $\Delta_t^a = d_a^{-1}\|a_t^{\mathrm{WDN}} - a_{t-1}^{\mathrm{WDN}}\|_1$ the mean absolute action change, where d_a is the action dimension. We use $[x]_+ = \max(x, 0)$. We define

$$\ell_t^{\mathrm{press}} = \frac{1}{|\mathcal{V}_{\mathrm{WDN}}|} \sum_{j \in \mathcal{V}_{\mathrm{WDN}}} \left[\left(\frac{p_{j,t}^{\mathrm{hyd}} - p^{\mathrm{tgt}}}{p^{\mathrm{tgt}}}\right)^2 + \left(\frac{[p^{\mathrm{min}} - p_{j,t}^{\mathrm{hyd}}]_+}{p^{\mathrm{min}}}\right)^2 + \left(\frac{[p_{j,t}^{\mathrm{hyd}} - p^{\mathrm{max}}]_+}{p^{\mathrm{max}}}\right)^2\right]. \tag{34a}$$

$$\ell_t^{\mathrm{sat}} = \frac{1}{|\mathcal{V}_{\mathrm{WDN}}|} \sum_{j \in \mathcal{V}_{\mathrm{WDN}}} \frac{[d_{j,t}^{\mathrm{dem}} - d_{j,t}^{\mathrm{del}}]_+}{d_{j,t}^{\mathrm{dem}}}. \tag{34b}$$

$$r_t = \exp\left(-\frac{\omega_e e_t^{\text{pump}} + \omega_p \ell_t^{\text{press}} + \omega_s \ell_t^{\text{sat}} + \omega_a \Delta_t^a}{\beta_r}\right) + b. \tag{34c}$$

Here, p^{tgt}, $p^{\min}$, and $p^{\max}$ are the target, minimum, and maximum pressures, respectively; $d_{j,t}^{\text{dem}}$ and $d_{j,t}^{\text{del}}$ denote requested and delivered demands at node j. The nonnegative weights balance the four components, $\beta_r > 0$ sets the reward scale, and b is an alive bonus. We set $\omega_e = 10^{-3}$, $\omega_p = 0.85$, $\omega_s = 0.5$, $\omega_a = 0.02$, $\beta_r = 1$, and $b = 0.02$. The reward applies at nonterminal steps; pressure, demand-satisfaction, or numerical-safety violations may terminate an episode.

E.2 Demand-Pattern Datasets

Static Demand. The static patterns are hourly water-consumption traces from university-community buildings collected during Fall 2018 (Bejarano et al., 2019). Each trace is identified by a two-letter building code and registered as an exogenous EPANET demand pattern for ordinary junctions.

Synthetic Demand. To complement the measured traces, we use **ChatGPT 5.6 sol** (**OpenAI**) to generate low-, medium-, and high-demand settings for node J7, denoted SynLow, SynMed, and SynHigh. Each profile follows a smooth bimodal daily pattern with water demand peaks around midday and in the evening. These fixed profiles are used exclusively in the demand-shift OOD scenarios.

E.3 Experimental Setup

Offline Data Collection. We use a well-trained Soft Actor-Critic (SAC) policy (Haarnoja et al., 2018) to collect the offline WDN dataset in `EPAGym`. The dataset contains 5,000 trajectories, each consisting of 240 control steps and yielding a return of approximately 170–200.

Method	Hyperparameters
`IQL`	Discount factor $\gamma = 0.99$; Learning rates $\eta_\pi = \eta_Q = \eta_V = 3 \times 10^{-6}$; Target-network update rate $\rho = 0.005$; Expectile $\alpha_{\text{IQL}} = 0.7$; Inverse temperature $\beta_{\text{IQL}} = 3$.
`FWM`	Window length $T_{\text{win}} = 12$; Context-history length 5; Flow-pretraining learning rate 2×10^{-4}; Discount factor 0.99; Condition-drop probability $p_{\text{drop}} = 0.2$; ODE solver: midpoint, 5 steps; Classifier-free guidance weight $\lambda_{\text{cfg}} = 2$.
`ATLA`	Perturbation radius $\varepsilon_{\text{ATLA}} = 0.025$; Perturbed-sample fraction 0.5; Actor adversarial weight $\lambda_{\text{adv}} = 0.1$; Adversary learning rate 3×10^{-4}; ℓ_2 regularization coefficient 0.02.
`RSC`	Importance threshold 0.5.
`ROMB`	Gaussian dynamics models used for rollouts 5; Dynamics-model learning rate 10^{-3}; Rollout horizon $H = 5$; GAE coefficient $\lambda_{\text{GAE}} = 0.95$; PPO clip range 0.2; Stackelberg correction coefficient 10^{-8}; Second-order horizon cap 10; Model constraint threshold 10.
`GURO-RL`	DRO budget $\epsilon = 1$; Lagrange multiplier $\mu = 10^{-6}$; Dual step size $\eta_\mu = 10^{-4}$; PPO clipping parameter $\epsilon_{\text{ppo}} = 0.4$; Flow fine-tuning learning rate 2.5×10^{-5}; Generated-future discount factor $\gamma = 0.99$; Window length $T_{\text{win}} = 12$; Rollout horizon $H = 5$; Inner optimization rounds $M = 2$; $\beta_Q = 0.4$.

Table 10: Hyperparameter settings for the WDN case study.

Training Protocol. All methods use the same SAC-collected offline WDN dataset and training seeds $\{0, 42, 3047\}$. Stage 1 pretrains the `IQL` backbone for 200 epochs with a batch size of 256, Stage 2 continues continue training on the same dataset for 50 epochs in Stage 2. All robust and generative variants perform one method-specific update every $F = 15$ offline training iterations and standard `IQL` updates otherwise. Flow models are pretrained for 2,000 epochs, while `ROMB` dynamics models are trained for 30 epochs. Table 10 summarizes the hyperparameters.

Table 11: Nominal and OOD settings for the WDN case study.

Scenario	Type	Assigned values
Nominal	Nominal	Nominal demand and parameters
Demand-pattern shift (J3, J4, J5, J6, J7)	Synthetic-Low	(BN, RA, GE, S2, SynLow)
	Synthetic-Medium	(CE, S3, LH, OO, SynMed)
	Synthetic-High	(DG, SN, FA, EB, SynHigh)
Pipe-property shift (Diameter, Roughness)	D-0.4 R-1.8	(0.4, 1.8)
	D-0.6 R-1.6	(0.6, 1.6)
	D-0.8 R-1.4	(0.8, 1.4)
Actuator-scale shift (P1, P5, P7, PUMP)	Scale-0.4	(0.4, 0.4, 0.4, 0.4)
	Scale-0.8	(0.8, 0.8, 0.8, 0.8)
	Scale-1.2	(1.2, 1.2, 1.2, 1.2)
	Scale-1.6	(1.6, 1.6, 1.6, 1.6)

Evaluation Protocol. This case study uses a separate domain-specific evaluation protocol. For every method, we evaluate its last five checkpoints for 10 episodes each, using an episode horizon of 240 control steps, and report episodic return. The nominal environment and three OOD shift families are summarized in Table 11. The nominal WDN uses static traces (EB, C4, JS, S3, BR) for (J3, J4, J5, J6, J7). Demand shifts replace these traces as specified in the table, with J7 using synthetic profiles of different scales. Pipe-property shifts scale pipe diameters and roughness relative to their nominal values. Actuator-scale shifts change the feasible PBV and pump-control ranges. We report mean unnormalized episodic returns as the performance metric.

E.4 Results

E.4.1 Main Results

Scenario	Type	Standard	Generative	Robust		Generative Robust	
		IQL	FWM	ATLA	RSC	ROMB	GURO-RL
Nominal	Nominal	198.8 ±1.4	204.4 ±2.1	199.7 ±1.4	200.6 ±2.1	201.8 ±1.7	205.7 ±0.5
Demand-pattern shift	Synthetic-Low	175.0 ±3.5	179.4 ±3.4	176.1 ±3.9	178.3 ±3.0	177.3 ±4.5	182.3 ±1.6
	Synthetic-Medium	130.2 ±2.5	134.4 ±3.7	129.6 ±1.9	131.2 ±2.8	133.3 ±5.7	133.7 ±0.8
	Synthetic-High	87.7 ±6.3	89.8 ±6.7	88.6 ±5.1	88.3 ±5.4	83.4 ±3.9	92.1 ±1.1
	Average	131.0 ±2.3	134.5 ±2.0	131.4 ±2.4	132.6 ±2.0	131.3 ±1.8	136.0 ±0.5
Pipe-property shift	D-0.4 R-1.8	174.3 ±5.3	175.1 ±3.5	175.4 ±4.2	173.3 ±4.4	173.6 ±3.0	176.6 ±0.4
	D-0.6 R-1.6	193.8 ±2.0	199.0 ±2.9	195.0 ±1.6	195.2 ±2.8	196.1 ±1.6	200.6 ±0.5
	D-0.8 R-1.4	197.7 ±1.5	203.5 ±2.2	198.8 ±1.4	199.5 ±2.1	200.8 ±1.8	204.8 ±0.5
	Average	188.6 ±2.6	192.5 ±2.9	189.7 ±2.0	189.3 ±3.0	190.2 ±1.6	194.0 ±0.5
Actuator-scale shift	Scale-0.4	97.3 ±11.5	105.5 ±12.9	99.3 ±9.7	97.7 ±9.6	91.0 ±7.9	109.3 ±2.9
	Scale-0.8	167.4 ±1.3	173.0 ±2.1	167.7 ±1.4	169.1 ±1.3	168.8 ±2.0	174.0 ±0.5
	Scale-1.2	178.2 ±14.4	199.0 ±16.0	178.8 ±13.8	182.6 ±14.4	174.1 ±15.1	208.4 ±2.0
	Scale-1.6	90.9 ±68.7	95.8 ±46.1	98.3 ±69.3	119.5 ±45.0	118.9 ±38.8	111.9 ±3.1
	Average	133.4 ±16.4	143.3 ±10.5	136.0 ±16.5	142.2 ±12.8	138.2 ±12.0	150.9 ±5.5
Average	OOD	149.3 ±6.1	155.5 ±5.0	150.7 ±6.0	153.5 ±5.4	151.7 ±5.2	159.4 ±2.3
	All	153.8 ±5.6	159.9 ±4.6	155.2 ±5.5	157.8 ±5.1	156.3 ±4.8	163.6 ±2.1

Table 12: WDN returns under nominal and OOD dynamics.

Table 12 shows that GURO−RL achieves the highest OOD and all-scenario mean returns of 159.4 and 163.6, respectively, as well as the highest nominal return. Among the ten OOD settings, FWM performs best in the demand-shift setting using Synthetic-Medium, and RSC performs best in Scale-1.6; GURO−RL achieves the highest mean return in the remaining eight settings. Its aggregate improvement therefore comes without an observed reduction in nominal performance.

WDN control involves coupled dynamics uncertainty: persistent changes in demand, pipe properties, and actuator capacity jointly affect successive hydraulic states, energy consumption, and rewards, motivating the non-rectangular ambiguity set used by GURO−RL. Relative to IQL, GURO−RL improves the all-scenario mean return by 9.8, accompanied by gains of 6.9 under nominal dynamics and 10.1 under OOD shifts. These improvements suggest that training on adverse multi-step continuations can strengthen robustness while preserving effective control under nominal conditions. Compared with FWM, GURO−RL increases the all-scenario mean return by 3.7, with improvements in both nominal and OOD returns. Since both methods use flow-based multi-step generation, this comparison is consistent with the benefit of adapting the generator to expose weaknesses of the current policy rather than keeping it frozen. The all-scenario gains over RSC and ATLA are 5.8 and 8.4, respectively. Their state or observation augmentation can reduce sensitivity to perturbed inputs, whereas GURO−RL explicitly searches for adverse multi-step state–reward continuations, which may better prepare the policy for persistent hydraulic changes. The 7.3 improvement in all-scenario mean return over ROMB may reflect the greater flexibility of our flow-based uncertainty set in representing high-dimensional state–reward distributions compared with explicit Gaussian transition models. Across these comparisons, the higher nominal returns indicate that the aggregate gains are not obtained at the expense of nominal performance. The flow-matching constraint limits departure from the offline distribution, while IPW and the relative value objective guide adversarial search toward policy-relevant actions and context-specific value degradation. Together, these components support the search for adverse futures while discouraging excessive deviations from the learned offline dynamics. The experimental results demonstrate that, beyond locomotion benchmarks, GURO−RL is also applicable to infrastructure-control simulations modeled on real-world environments.

E.4.2 Sensitivity to the horizon

Scenario	Type	$H=1$	$H=3$	$H=5$	$H=7$	$H=9$
Nominal	Nominal	201.1 ±0.6	203.8 ±0.7	205.7 ±0.5	204.5 ±0.7	204.1 ±0.8
Demand-pattern shift	Synthetic-Low	174.5 ±2.9	169.8 ±1.9	182.3 ±1.6	179.5 ±1.4	181.5 ±1.6
	Synthetic-Medium	127.5 ±2.4	126.8 ±0.4	133.7 ±0.8	132.6 ±0.7	133.1 ±0.8
	Synthetic-High	85.6 ±4.1	87.5 ±0.8	92.1 ±1.1	90.4 ±1.3	90.1 ±0.9
	Average	129.2 ±3.1	128.0 ±1.0	136.0 ±1.2	134.2 ±1.1	134.9 ±1.1
Pipe-property shift	D-0.4 R-1.8	176.6 ±2.5	176.3 ±0.6	176.6 ±0.4	175.4 ±0.5	174.1 ±0.5
	D-0.6 R-1.6	196.0 ±0.5	198.7 ±0.6	200.6 ±0.5	200.3 ±0.7	199.0 ±0.7
	D-0.8 R-1.4	199.8 ±0.7	202.7 ±0.7	204.8 ±0.5	205.5 ±0.7	206.3 ±0.7
	Average	190.8 ±1.2	192.6 ±0.6	194.0 ±0.5	193.7 ±0.6	193.1 ±0.6
Actuator-scale shift	Scale-0.4	100.8 ±7.6	107.3 ±2.4	109.3 ±2.9	115.4 ±2.6	116.9 ±1.7
	Scale-0.8	167.8 ±1.2	170.1 ±0.7	174.0 ±0.5	175.7 ±0.9	176.5 ±1.0
	Scale-1.2	165.5 ±4.6	177.7 ±7.6	208.4 ±2.0	206.0 ±2.1	205.5 ±2.9
	Scale-1.6	96.8 ±7.0	53.0 ±3.7	111.9 ±3.1	41.9 ±2.0	26.3 ±7.3
	Average	132.7 ±5.1	127.0 ±3.6	150.9 ±2.1	134.8 ±1.9	131.3 ±3.2
Average	OOD	149.1 ±3.4	147.0 ±1.9	159.4 ±1.3	152.3 ±1.3	150.9 ±1.8
	All	153.8 ±3.1	152.2 ±1.8	163.6 ±1.3	157.0 ±1.2	155.8 ±1.7

Table 13: Sensitivity of WDN performance to the horizon H.

Table 13 varies the policy-scoring horizon while holding the remaining WDN settings fixed. The intermediate choice $H = 5$ attains the highest nominal return and the highest Demand, Pipe Property, Actuator Scale, OOD, and all-scenario averages. Longer horizons improve a few individual shifts, but these gains do not persist in aggregate. This pattern may also depend on generative-model capacity: we use the same flow-model architecture for MuJoCo and WDN, whose fixed capacity may limit its ability to accurately capture longer-range dependencies. We therefore use $H = 5$ for the WDN main results.

F NOTATION

Table 14: Core and backbone notation. Other proof-, baseline-, and WDN-specific symbols are defined locally.

Notation	Description
Offline RL and distributional robustness	
$\mathcal{M}$	Discounted Markov decision process
$\mathcal{S}$, $\mathcal{A}$, γ	State space, action space, and discount factor
s_t, a_t, r_t	State, action, and reward at environment step t
π, π_b, π_0	Current policy, data-collection policy, and pretrained policy used as a surrogate for π_b
$\mathcal{D}$, N	Fixed offline trajectory dataset and number of trajectories
V_ν^π, Q_ν^π	State and action values under policy π and kernel ν
$\mathcal{U}$, $\mathcal{U}_{\mathrm{G}}$	General uncertainty set and flow-parameterized uncertainty set
$P_\circ$, ϵ	Nominal conditional trajectory distribution and ambiguity budget
W_2, ϵ_{W}	2-Wasserstein distance and budget on its averaged conditional square
Conditional flow models	
H, T_{win}	Rollout/value-target horizon and flow-pretraining window length
τ	Joint H-step reward–state trajectory
y, Y	Initial state–action condition and its random-variable form
g_t	Discounted reward sum over the trajectory window; additional condition used only by `FWM`
θ, θ_{old}	Current flow parameters and fixed parameters used to sample trajectories for the surrogate update
X_θ, $P_\theta(\cdot \mid y)$	Conditional trajectory sampler and its induced distribution
p_0, p_1	Flow source distribution (standard Gaussian) and target data distribution
X_0, X_1, X_k	Source, target, and interpolated samples
k, U	Continuous flow time in $[0,1]$ and linear-path target velocity
ζ, p_{drop}, $\widetilde{Y}$	Bernoulli dropout variable, dropout probability, and retained or empty condition
u_θ, u_θ^{gfm}, $u^\star$	Learned, classifier-free-guided, and ideal conditional velocity fields
λ_{gfm}, L_{gfm}, C_{gfm}	Guidance weight, guided-field Lipschitz constant, and residual-bound factor in Proposition 4.1
J_{CFM}, J_{GFM}	Conditional and guided flow-matching objectives
IPW and adversarial optimization	
w_π	Inverse propensity weight at an observed state–action pair
β	Softmax inverse temperature in the approximate IPW formula
β_Q	Generated-Q weighting strength in the experimental configurations
$E(\tau)$	Discounted trajectory return with a terminal value bootstrap
$\tau_{\mathcal{D}}(s,a)$, $\Delta_E(\tau)$	Offline reference trajectory and generated value minus the mean reference value for the same (s,a)
$A(\tau)$	Negative IPW-weighted relative value used in the adversarial surrogate
$\widehat{J}_{\mathrm{GFM}}$, N_ϵ	Sample-wise flow residual estimate and number of sampled noise–time pairs used to compute it
$r_\theta(\tau)$	Surrogate ratio derived from the old and current flow residual estimates
ϵ_{ppo}, L_{ppo}	PPO clipping radius and clipped adversarial objective
μ, η_μ	Nonnegative Lagrange multiplier and projected dual-update step size

Continued on next page

Table 14 (continued)

Notation	Description
Backbone updates (Appendix B)	
$y_{\mathcal{D},i} = (s_{0,i}, a_{0,i})$	Real initial state–action pair for batch sample i
$\tau^{\mathrm{adv}}_{\theta,i}$	Generated adversarial continuation for batch sample i
$\mathcal{B}$, B	Adversarial batch with its real initial contexts, and batch size
ϕ, ψ, $\bar{\phi}$	Critic, value-network, and target-critic parameters
Q^{π}_{ϕ}, V^{π}_{ψ}, $Q^{\pi}_{\bar{\phi}}$	Learned critic, value network, and target critic
ϕ_j, $\bar{\phi}_j$, $j = 1, 2$	Individual BCQ critic parameters and their target copies
ϑ_π	IQL/CQL actor parameters, with $\pi = \pi_{\vartheta_\pi}$
α_{IQL}, β_{IQL}	IQL expectile level and advantage-weight inverse temperature
$\ell_{\alpha_{\mathrm{IQL}}}$	Asymmetric squared loss for IQL expectile regression
α_{CQL}, α_{ent}	CQL conservative-penalty and entropy coefficients
$G_{\mathrm{VAE}}(\cdot \mid s)$	BCQ conditional behavior VAE for dataset actions
$\omega = (\omega_1, \omega_2)$	BCQ VAE encoder and decoder parameters
Enc_{ω_1}, Dec_{ω_2}	BCQ VAE encoder and decoder
μ_z, σ_z	Mean and elementwise standard deviation of the encoded latent Gaussian
z, z^{rec}, d_z	VAE latent variable, reconstruction latent draw, and latent dimension
$\mathcal{Z}$, $z^{(m)}$	Cached latent candidate set and its m-th draw
$\vartheta_{\mathrm{pert}}$, $\bar{\vartheta}_{\mathrm{pert}}$	BCQ perturbation parameters and their target copy
g_{pert}, $\bar{g}_{\mathrm{pert}}$	BCQ adjusted-action map and its target copy
$\Delta_{\vartheta_{\mathrm{pert}}}$	Bounded action residual in the BCQ perturbation model
Φ_{BCQ}, λ_{BCQ}, N_{cand}	BCQ perturbation bound, double-Q mixture coefficient, and candidate count
$\mathcal{C}_{\mathrm{BCQ}}(s)$	Candidate actions produced by the VAE and target perturbation model
$a^{(m)}$	Action sampled from the BCQ behavior VAE for policy extraction
$\mathcal{L}^{\mathrm{rob}}_Q$, $\mathcal{L}_V$, $\mathcal{L}_\pi$	Robust critic, IQL value, and IQL/CQL policy losses
$\mathcal{L}_{\mathrm{VAE}}$, $\mathcal{L}_{\mathrm{pert}}$	BCQ VAE and perturbation losses
η_Q, η_V, η_π	Critic, value-network, and policy learning rates
η_{VAE}, η_{pert}	BCQ VAE and perturbation learning rates
$\mathbb{E}_{\mathcal{B}}$, $\mathbb{E}_{\mathcal{B},\mathcal{Z}}$	Batch empirical average, and average also over candidate latent draws
Training protocol and evaluation	
M, F, ρ	Inner update count, robust-update interval, and target-network update rate
$\bar{D}$, $\mathcal{I}$	Mean paired state-distribution W_2 distance and comparison index set over methods, seeds, and OOD settings
$\bar{R}$, $R_{\min}$, $R_{\max}$	Mean episodic return and reference returns for score normalization