DROPS: ROBUST PARETO SET LEARNING UNDER PREFERENCE DRIFT FOR MULTI-OBJECTIVE ROBOTIC CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-objective Robotic Control is increasingly important in real-world applications, with Multi-Objective Reinforcement Learning (MORL) providing a promising approach for learning policies over conflicting objectives. However, existing MORL methods typically assume clean observations and fixed environment dynamics. These assumptions often fail in practice, where uncertainty can cause severe ***preference drift***, substantially undermining the robustness of learned control policies. In this paper, we propose **DROPS**, a Preference-Drift-Aware approach for learning a Robust Pareto Set of control policies against uncertainty. Specifically, DROPS constructs ***(i) a drift-aware trust region*** for each preference, introduces ***(ii) hierarchical adversarial perturbations*** to establish a degradation bound with a policy-independent coefficient, and leverages ***(iii) contrastive drift alignment*** to recover robust preference-to-policy mappings. Central to our method is strengthening the robustness of the learned policy Pareto Set by tightening the degradation bound and aligning drifted preferences, enabling DROPS to learn a Pareto Set robust to preference drift. Experiments on five continuous robotic-control tasks, together with a discrete benchmark, show its consistently superior performance in obtaining robust control policies.

1 INTRODUCTION

Multi-objective Robotic Control has become increasingly important across a wide range of real-world tasks (Tang et al., 2025), where robotic agents are often required to simultaneously account for multiple, potentially conflicting objectives. A promising paradigm is to formulate such control problems within the paradigm of Multi-Objective Reinforcement Learning (MORL), which enables agents to learn policies that capture different trade-offs among objectives. Multi-objective problems result in a set of Pareto-optimal policies (Pareto Set) and MORL methods seek to learn a Pareto Set comprising these preference-conditioned policies, with each policy tailored to a specific preference over the objectives (Basaklar et al., 2022).

Existing MORL methods achieve impressive performance in uncertainty-free settings, assuming reliable observations and stationary environments. However, real-world scenarios are inevitably subject to natural uncertainty, which can substantially undermine their robustness. As shown in Figure 1, the Pareto Set of motion policies learned in uncertainty-free settings for a quadruped robot trading off speed and energy, often fails in real-world deployment due to uncertainties (Singi et al., 2024; Ghasemipour et al., 2022). Such uncertainties can be categorized as ***(i) observation perturbation*** from sensor noise or

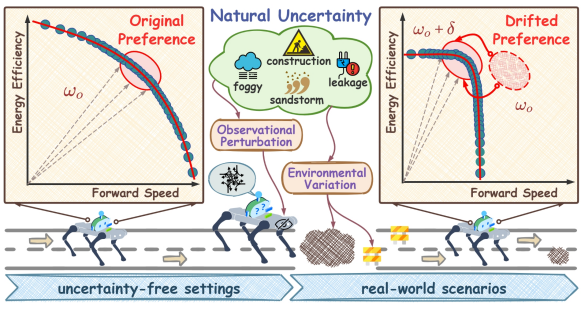


Figure 1: Comparison of robotic control policies. Maintaining a trade-off strategy may lead to ***preference drift*** as the Pareto Set evolves under uncertainty.

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.

actuation errors, and ***(ii) environmental variation*** due to joint damping variation or contact-friction shifts. From the observational perspective, unexpected disturbances in sensing conditions may induce state perturbations with unknown distributions (Singi et al., 2024; Lockwood & Si, 2022), thereby degrading preference-conditioned policies. Although prior works attempt to predict these distributions for proactive adaptation (Wei et al., 2023; Yang et al., 2024b), their practical robustness is often limited by the difficulty of accurately modeling uncertainty under ambiguous conditions. From the environmental perspective, unpredictable changes in the environment can drive the system into out-of-distribution states that exceed predefined operating bounds or even lack feedback (Di Langosco et al., 2022; An et al., 2021). Several approaches enhance robustness through specific mechanisms (Mo et al., 2024; He et al., 2024), yet they overlook the fact that the realized trade-off may become inconsistent with the requested preference under uncertainty.

Uncertainties can reshape the robot's Pareto Set by forcing drifts in the mapping from preferences to policies, thereby changing the resulting preference-conditioned trade-offs, as illustrated in Figure 1. We refer to this structural change in trade-offs caused by uncertainties as ***preference drift***, which prevents existing methods from reliably preserving the intended Pareto trade-offs under uncertainty. The challenges of adapting to this unexplored preference drift stem from two main factors:

- **How to jointly handle both uncertainties to enhance robustness?**Treating them separately is often insufficient, as their coupled effects can still disrupt learned policies and shift the trade-off structure. Only a few existing studies consider both uncertainties jointly, yet they do not explicitly quantify the relationship between uncertainty modeling and robustness.
- **How to achieve robust policy generation in the presence of preference drift?** The reshaped underlying trade-off structure can render the preference–policy mapping misaligned or even invalid; consequently, a policy generated for a given preference may no longer realize the intended trade-offs. To the best of our knowledge, no existing work explicitly targets preference drift or provides dedicated mechanisms.

In this light, we propose Preference-Drift-Aware Robust Pareto Set Learning (**DROPS**), which extends adversarial perturbations and integrates them with contrastive learning for MORL, aiming to learn a robust Pareto Set. DROPS adapts to preference drift at different stages of Pareto Set Learning (PSL), improving robustness in the policy network for reliable rollouts and in the hypernetwork for robust preference-conditioned policy generation. To tackle these challenges, we ***(i) construct a drift-aware trust region*** to characterize the robustness neighborhood of each preference; ***(ii) develop hierarchical adversarial perturbations at step- and episode-level*** to jointly capture observational and environmental uncertainties, with their effects theoretically bounded independent of policy structures; and ***(iii) steer drifted-preference policies toward the trust region*** via contrastive learning. The main contributions are summarized as follows:

- **New Problem Identification.** We highlight ***preference drift*** as an underexplored factor in robustness failures in Multi-Objective Robotic Control, and propose DROPS to address it under observational and environmental uncertainties.
- **New approach to Robustness.** DROPS strengthens Pareto Set robustness by tightening the degradation bound and by leveraging contrastive learning for drift alignment to steer drifted preferences towards the drift-aware trust region.
- **Robust Robotic Control Performance.** Experiments on five MO-MuJoCo robotic-control tasks demonstrate that DROPS improves Pareto quality and preference robustness under both uncertainty, with complementary results on a discrete benchmark.

2 Problem Formulation

Pareto Set Learning. Multi-objective robotic control problems can be formulated as multi-objective sequential decision-making problems and addressed within the MORL paradigm, where a policy π is evaluated by an m-dimensional return vector $\mathcal{J}(\pi) = \big(J_1(\pi), \ldots, J_m(\pi)\big) \in \mathbb{R}^m$. Pareto Set Learning is a MORL framework that learns the Pareto Set as a continuous preference-conditioned structure (Xu et al., 2020). Let $\Delta^{m-1} = \big\{\omega \in \mathbb{R}^m_+ \mid \mathbf{1}^\top \omega = 1\big\}$ denote the preference simplex. A PSL method maps a preference $\omega \in \Delta^{m-1}$ to a preference-conditioned policy π_ω and subsequently to its realized multi-objective return. In practical deployment, the policy may operate

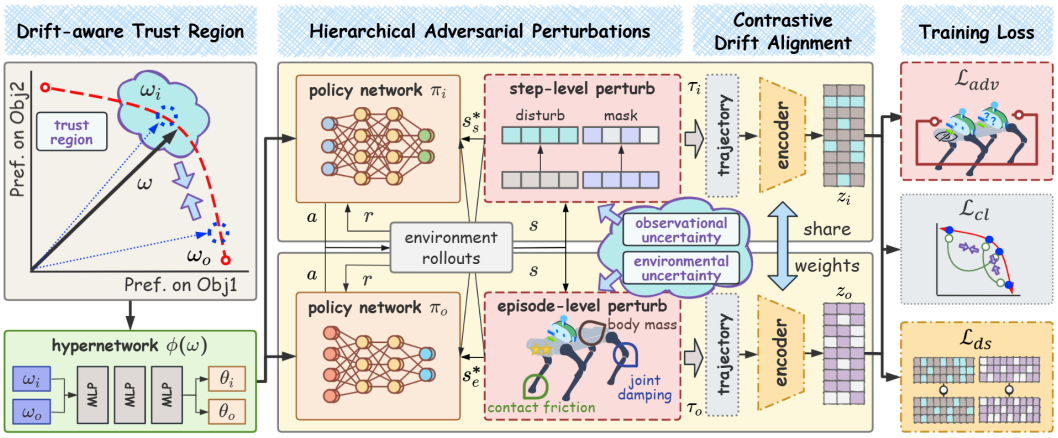


Figure 2: Workflow of the proposed DROPS. DROPS constructs a Drift-aware Trust Region to generate behaviorally near and distant preference samples, improves preference-conditioned policy robustness through Hierarchical Adversarial Perturbations ($\mathcal{L}_{adv}$), corrects uncertainty-induced preference correspondence through Contrastive Drift Alignment ($\mathcal{L}_{cl}$), and preserves global Pareto diversity through $\mathcal{L}_{ds}$ to prevent degenerate policy collapse.

under both observational and environmental uncertainties. We denote an uncertainty realization by $u = (\nu, \xi)$, where ν represents observation corruption and ξ represents an environment configuration. Accordingly, uncertainty induces a family of partially observable environments, where latent dynamics follow an uncertainty-dependent kernel p_ξ and observations may be corrupted by ν.

Preference Drift. In this work, we consider reactive preference-conditioned policies rather than solving the general belief state POMDP. For a preference ω, let $F_0(\omega) = \mathcal{J}_0(\omega)$ denote its nominal return and let $F_u(\omega) = \mathcal{J}_u(\omega)$ denote its expected return under uncertainty u. Although uncertainty may moderately affect the Pareto Set, it can alter the correspondence between a requested preference and its realized trade-off. We define this phenomenon as

Definition 1 (Preference drift). *For any preference $\omega \in \Delta^{m-1}$ and uncertainty realization u, we define the set of nominal preferences whose returns best match the uncertain return $F_u(\omega)$ as $\Omega_u^\star(\omega) \triangleq \arg\min_{\omega' \in \Delta^{m-1}} \|F_0(\omega') - F_u(\omega)\|_2$. The magnitude of preference drift is defined as $d_{\mathrm{PD}}(\omega; u) \triangleq \mathrm{dist}\,(\omega, \Omega_u^\star(\omega)) = \min_{\bar{\omega} \in \Omega_u^\star(\omega)} \|\omega - \bar{\omega}\|_2$. Preference drift occurs when $d_{\mathrm{PD}}(\omega; u) > 0$, equivalently when $\omega \notin \Omega_u^\star(\omega)$.*

Intuitively, $\Omega_u^\star(\omega)$ identifies the nominal preferences whose returns are most consistent with the return realized by ω under uncertainty. Therefore, $d_{\mathrm{PD}}(\omega; u)$ measures how far the realized trade-off has moved in the preference space rather than merely measuring degradation in the objective space. Since preference drift is a property of the preference-to-performance correspondence rather than of an unordered solution set alone, We use the term ***robust Pareto Set*** to refer to a preference-conditioned Pareto Set. For a hypernetwork ϕ, let $\mathcal{P}_\phi = \{(\omega, \pi_{\phi(\omega)}) \mid \omega \in \Delta^{m-1}\}$. We characterize its preference robustness under an uncertainty set $\mathcal{U}$ by $\mathcal{R}_{\mathrm{PD}}(\phi; \mathcal{U}) \triangleq \sup_{u \in \mathcal{U}} \mathbb{E}_{\omega \sim q(\omega)} [d_{\mathrm{PD}}(\omega; u)]$, where $q(\omega)$ denotes the preference distribution. A smaller $\mathcal{R}_{\mathrm{PD}}$ indicates that the learned Pareto Set better preserves the nominal preference-to-performance correspondence under uncertainty. **Our objective is not only to preserve Pareto quality, but also to explicitly reduce preference drift.**

3 Proposed Method

As shown in Figure 2, DROPS trains a hypernetwork that conditions on preferences sampled inside and outside a drift-aware trust region. The three components address complementary aspects of preference drift: ***(i) Drift-Aware Trust Region*** provides behaviorally controlled reference and drift

candidate preferences for training. ***(ii) Hierarchical Adversarial Perturbations*** expose preference-conditioned policies to observational and environmental uncertainties and reduce the resulting return displacement. Because preference drift arises from changes in the realized preference-to-return correspondence, suppressing return displacement provides the first level of protection against drift. ***(iii) Contrastive Drift Alignment*** explicitly regularizes the hypernetwork mapping by correcting drift candidate behaviors toward their nominal correspondence; a diversity loss prevents this alignment from being trivially achieved by mapping different preferences to nearly identical policies.

3.1 Drift-aware Trust Region Construction

Trust Region. To construct informative samples for preference drift adaptation, we introduce a drift-aware trust region around each anchor preference. Importantly, the trust region serves as a training mechanism for constructing behaviorally near reference preferences and distant drift candidates. Let π_ω denote the policy produced by the hypernetwork conditioned on ω. The trust region around ω is defined by constraining the divergence between preference conditioned policies:

$$T_\omega(\rho) \triangleq \left\{ \omega' \in \Delta^{m-1} \;\middle|\; \bar{D}(\pi_{\omega'} \| \pi_\omega) \le \rho \right\}, \tag{1}$$

where ρ is a predefined radius and $\bar{D}(\pi_{\omega'} \| \pi_\omega)$ denotes the expected KL divergence between $\pi_{\omega'}(\cdot|s)$ and $\pi_\omega(\cdot|s)$ under the reference state distribution. We assume that the preference-to-policy mapping is locally Lipschitz continuous, namely, there exists $L > 0$ such that, $\forall \omega, \omega' \in \Delta^{m-1}$, $\bar{D}(\pi_{\omega'} \| \pi_\omega) \le L \|\omega' - \omega\|_2$. Specifically, choosing a perturbation ϵ satisfying $\|\epsilon\|_2 \le \frac{\rho}{L}$ and $\omega + \epsilon \in \Delta^{m-1}$, guarantees that $\omega_i = \omega + \epsilon \in T_\omega(\rho)$.

We sample candidate preferences $\tilde{\omega}_o = \omega + \delta$ with $\|\delta\|_2 > \|\epsilon\|_2$ and explicitly retain those satisfying $\bar{D}\left(\pi_{\tilde{\omega}_o} \| \pi_\omega\right) > \rho$. The accepted preference is denoted by ω_o and satisfies $\omega_o \notin T_\omega(\rho)$ by construction. Hence, ω_i serves as a behaviorally local reference, whereas ω_o serves as a hard drift candidate. These samples do not replace the preference drift definition and provide tractable training surrogates for learning a mapping robust to uncertainty-induced preference correspondence changes.

3.2 Hierarchical Adversarial Perturbations

Given the drift-related preferences ω_i and ω_o, the hypernetwork produces the corresponding policy parameters θ_i and θ_o. We apply hierarchical adversarial perturbations to these policies to reduce uncertainty-induced return displacement, which constitutes a primary source of preference drift.

Perturbation Set. We distinguish observational uncertainty from environmental uncertainty. Let s denote the latent environment state. A step-level perturbation ν corrupts the observation received by the policy, while an episode-level perturbation ξ modifies an environment configuration and remains fixed within an episode. We define the uncertainty set as

$$\mathcal{U} \triangleq \{u = (\nu, \xi) \mid \|\nu\|_\infty \le \zeta,\ d_\Xi(\xi, \xi_0) \le \eta\}, \tag{2}$$

where ζ and η control the severity of observational and environmental uncertainties, d_Ξ denotes a normalized distance and ξ_0 denotes the nominal environment configuration. An uncertainty realization $u = (\nu, \xi)$ induces a partially observable environment $\mathcal{M}_u = (\mathcal{S}, \mathcal{O}, \mathcal{A}, R, p_\xi, O_\nu, \gamma)$, where p_ξ denotes the environment transition kernel and O_ν determines the observation received by the policy. We consider a reactive policy and denote its uncertainty-induced action distribution over the latent state by $\Pi_{\omega,u}(\cdot|s) \triangleq \pi_\omega\big(\cdot \mid O_\nu(s)\big)$. This formulation separates observation corruption from environment variation and does not require the policy to solve the general belief state POMDP.

Degradation Bound. We next establish how uncertainty-induced policy and environment variations affect the realized multi-objective return. For an uncertainty realization $u = (\nu, \xi)$, we define

$$\delta_\pi(\omega, u) \triangleq \max_s D_{\mathrm{TV}}\left(\pi_\omega(\cdot|s), \Pi_{\omega,u}(\cdot|s)\right), \qquad \delta_p(u) \triangleq \max_{s,a} D_{\mathrm{TV}}\left(p_{\xi_0}(\cdot|s,a), p_\xi(\cdot|s,a)\right). \tag{3}$$

Assume that each objective reward is bounded by $|R_k| \le R_{\max}$. Using a standard performance difference decomposition (Achiam et al., 2017), the degradation of objectives can be bounded as

$$|J_{0,k}(\omega) - J_{u,k}(\omega)| \le \kappa \left[\delta_\pi(\omega, u) + \delta_p(u)\right], \tag{4}$$

where the coefficient κ is independent of the specific policy parameterization, while δ_π captures the policy-dependent sensitivity to uncertainty. The return degradation directly characterizes a source of preference drift. Assume that the nominal return map is locally inverse stable in the considered preference region, namely, there exists $\mu > 0$ such that $\|F_0(\omega_1) - F_0(\omega_2)\|_2 \geq \mu \|\omega_1 - \omega_2\|_2$. Then, the preference drift satisfies $d_{\mathrm{PD}}(\omega; u) \leq \frac{2\sqrt{m}\kappa}{\mu} [\delta_\pi(\omega, u) + \delta_p(u)]$, which establishes an explicit connection between robustness and preference drift: **reducing uncertainty-induced policy and transition deviations tightens an upper bound on preference drift**.

Adversarial Loss. The degradation bound motivates minimizing the effect of both levels of uncertainty during training. Since total variation distance is difficult to optimize directly, we use Pinsker's inequality $D_{\mathrm{TV}}^2(P, Q) \leq \frac{1}{2} D_{\mathrm{KL}}(P \| Q)$ and optimize a worst-case adversarial objective:

$$\mathcal{L}_{adv}(\Theta) = \max_{u=(\nu,\xi)\in\mathcal{U}} \left[\mathcal{L}_{RL}(\Theta; u) + \lambda \max_s D_{\mathrm{KL}} \left(\pi_\Theta(\cdot|s) \| \Pi_{\Theta,u}(\cdot|s) \right) \right], \tag{5}$$

where $\Theta \in \{\theta_i, \theta_o\}$, $\mathcal{L}_{RL}$ denotes the loss optimized by the underlying RL algorithm, and λ controls the robustness regularization. The episode-level adversary searches for challenging environment configurations ξ, while the step-level adversary generates observation perturbations ν. Therefore, $\mathcal{L}_{adv}$ is not only used to improve conventional policy robustness; it also suppresses an important source of preference drift by limiting uncertainty-induced return displacement.

3.3 Contrastive Drift Alignment

Although $\mathcal{L}_{adv}$ reduces the return displacement that gives rise to preference drift, controlling performance degradation alone does not explicitly preserve the correspondence between preferences and their realized trade-offs. We therefore introduce contrastive drift alignment to directly regularize the preference conditioned mapping represented by the hypernetwork.

Drift-related Trajectory Encoding. For a given anchor preference ω, we construct a triplet $(\omega, \omega_i, \omega_o)$, where $\omega_i \in T_\omega(\rho)$ is a behaviorally local reference preference and $\omega_o \notin T_\omega(\rho)$ is a hard drift candidate. The three preferences are passed through the hypernetwork to generate their corresponding policy networks and are rolled out under observational and environmental uncertainties to obtain (τ, τ_i, τ_o). A shared trajectory encoder $\psi(\cdot)$ then produces (z, z_i, z_o). These embeddings provide a differentiable behavioral representation of the preference-to-policy correspondence. They are not intended to collapse heterogeneous trajectories into a single representation. Instead, they provide a surrogate space in which DROPS can regulate uncertainty-induced correspondence changes that would otherwise require repeatedly solving drift during training.

Drift Alignment. Preference drift indicates that the behavior realized from a requested preference becomes more consistent with the nominal behavior associated with another preference. Accordingly, we use the trajectory representation to correct such correspondence changes. For an anchor preference ω, the out-of-region sample ω_o is treated as a drift candidate whose uncertain behavior is encouraged to remain aligned with the anchor behavior. In contrast, nearby in-region preferences ω_i retain distinct representations so that correcting drift does not eliminate meaningful preference-dependent variations. Therefore, contrastive drift alignment simultaneously encourages robustness of the anchor correspondence and preserves local discrimination across the preference space.

Contrastive Loss. We formulate the contrastive objective using InfoNCE (Gutmann & Hyvärinen, 2010). Given an anchor embedding z, an out-of-region drift candidate embedding z_o, and embeddings $\{z_i^j\}_{j=1}^K$ from K in-region reference preferences, we define

$$\mathcal{L}_{cl} = -\log \frac{\exp\left(\mathcal{S}(z, z_o)/t\right)}{\exp\left(\mathcal{S}(z, z_o)/t\right) + \sum_{j=1}^K \exp\left(\mathcal{S}(z, z_i^j)/t\right)}, \tag{6}$$

where t denotes the temperature parameter and $\mathcal{S}(\cdot, \cdot)$ denotes cosine similarity. Minimizing $\mathcal{L}_{cl}$ increases the behavioral consistency between the anchor and its drift candidate while maintaining discrimination among nearby preferences. Hence, $\mathcal{L}_{cl}$ complements $\mathcal{L}_{adv}$: the latter limits the return displacement that induces preference drift, whereas the former explicitly regularizes the preference-to-policy correspondence once such drift tendencies arise.

3.4 Diversity Promoting for the Pareto Set

Although $\mathcal{L}_{cl}$ promotes preference correspondence robustness, contrastive alignment alone may admit a degenerate solution in which the hypernetwork generates highly similar policy parameters for different preferences. Such a collapse could reduce behavioral discrepancy without genuinely preserving a meaningful Pareto Set. Therefore, global diversity is necessary to ensure that preference robustness is not achieved at the cost of preference discrimination and Pareto coverage.

Embedding Diversity. The contrastive loss encourages $\mathcal{S}\big(\mathcal{Z}(\phi(\omega)), \mathcal{Z}(\phi(\omega + \delta))\big) \gg \mathcal{S}\big(\mathcal{Z}(\phi(\omega)), \mathcal{Z}(\phi(\omega + \epsilon))\big)$, where $\phi(\cdot)$ denotes the policy parameters generated by the hypernetwork and $\mathcal{Z}(\cdot)$ denotes their trajectory embedding. However, if ϕ maps a large range of preferences into a compact parameter region, the contrastive objective can be partially satisfied through a collapsed representation. From the perspective of the hypernetwork, such a degenerate mapping can be expressed as $\phi(\omega) \approx \phi(\omega + \delta) \approx \phi(\omega + \epsilon)$. This solution is undesirable because a robust preference-to-policy mapping must simultaneously satisfy two requirements: it should resist uncertainty-induced preference drift while retaining meaningful differences among policies associated with distinct preferences. Therefore, the diversity regularizer $\mathcal{L}_{ds}$ complements $\mathcal{L}_{adv}$ and $\mathcal{L}_{cl}$ by preventing trivial correspondence preservation through policy collapse.

Diversity Loss. To avoid behavior collapse, we introduce a diversity loss $\mathcal{L}_{ds}$ based on maximizing the entropy of the trajectory embedding distribution. Specifically, we approximate the true density function of embeddings z using Kernel Density Estimation. Given a mini-batch of trajectory embeddings $\{z_{i,j}\}_{i,j=1}^{N}$ obtained from the encoder $\psi(\{\tau_{i,j}\}_{i,j=1}^{N})$, we estimate the density at each point using a Gaussian kernel:

$$\mathcal{L}_{ds} = \frac{1}{N}\sum_{i=1}^{N} \log\left(\frac{1}{N}\sum_{j=1}^{N} e^{-\frac{\|z_i - z_j\|^2}{2\sigma^2}}\right), \tag{7}$$

where N is the batch size, σ is the kernel bandwidth, and $\|\cdot\|^2$ denotes the Euclidean norm in the latent space. Minimizing $\mathcal{L}_{ds}$ encourages the learned embeddings $z_{i,j}$ to spread across the latent space, rather than collapsing to a narrow manifold.

4 Experimental Study

4.1 Experimental Setup

Benchmark Problems. We center our evaluation on five continuous robotic-control tasks from MO-MuJoCo (Xu et al., 2020). To examine whether the proposed preference-drift mechanism generalizes beyond robotic dynamics, we additionally use the discrete FTN benchmark (Yang et al., 2019) as a complementary test. We introduce step-level uncertainty in observation settings and episode-level uncertainty in environment configurations. Details of the benchmarks and the uncertainty design are provided in **Appendix A**.

Competing Methods. We compare DROPS against five state-of-the-art methods: PD-MORL (Basaklar et al., 2022), PG-MORL (Xu et al., 2020), RMORL (He et al., 2024), Hyper-MORL (Shu et al., 2024), and PSL-MORL (Liu et al., 2025). To maintain fairness, we fine-tuned the default hyperparameter settings for all methods. For more details, please refer to **Appendix B**.

Evaluation Metrics. We employ three metrics: (i) Hypervolume (HV) (Shang et al., 2020); (ii) Spacing (SP) (Jiang et al., 2014); and (iii) Robustness (U). A larger HV value indicates a better Pareto Set; a lower SP value reflects a better diversity of solutions; and a lower U indicates a lower degree of preference drift under uncertainty. The definition of U is provided in **Appendix B.2**.

4.2 Experimental Results

How does the proposed DROPS perform compared with SOTAs under uncertainties? We evaluate both the quality and robustness of the Pareto Set. For fairness, we run each method with seven random seeds and report the results in Table 1 and Table 2.

Problem	Metrics	RMORL	Hyper-MORL	PD-MORL	PSL-MORL	PG-MORL	**DROPS**
MO-Swimmer-v2	HV($\times 10^3$)	$4.22_{\pm 0.16}$	$2.19_{\pm 0.11}$	$3.24_{\pm 0.18}$	$3.01_{\pm 0.10}$	$1.58_{\pm 0.15}$	$\mathbf{4.97}_{\pm 0.08}$
	SP($\times 10^1$)	$\mathbf{0.73}_{\pm 0.18}$	$2.23_{\pm 0.27}$	$3.26_{\pm 0.21}$	$1.91_{\pm 0.05}$	$2.50_{\pm 0.21}$	$0.77_{\pm 0.36}$
	U	$0.30_{\pm 0.09}$	$0.21_{\pm 0.15}$	$0.29_{\pm 0.02}$	$0.22_{\pm 0.17}$	$0.35_{\pm 0.13}$	$\mathbf{0.16}_{\pm 0.12}$
MO-HalfCheetah-v2	HV($\times 10^5$)	$1.41_{\pm 0.26}$	$1.18_{\pm 0.14}$	$0.24_{\pm 0.14}$	$0.22_{\pm 0.10}$	$0.58_{\pm 0.35}$	$\mathbf{2.45}_{\pm 0.23}$
	SP($\times 10^1$)	$\mathbf{0.82}_{\pm 0.12}$	$1.27_{\pm 0.22}$	$2.22_{\pm 0.29}$	$0.93_{\pm 0.19}$	$2.21_{\pm 0.19}$	$0.90_{\pm 0.05}$
	U	$0.20_{\pm 0.07}$	$0.31_{\pm 0.19}$	$0.22_{\pm 0.04}$	$0.25_{\pm 0.12}$	$0.37_{\pm 0.16}$	$\mathbf{0.18}_{\pm 0.13}$
MO-Walker2d-v2	HV($\times 10^5$)	$3.65_{\pm 0.30}$	$1.98_{\pm 0.43}$	$1.31_{\pm 0.45}$	$3.22_{\pm 0.28}$	$3.16_{\pm 0.14}$	$\mathbf{4.05}_{\pm 0.18}$
	SP($\times 10^1$)	$1.09_{\pm 0.13}$	$1.27_{\pm 0.33}$	$2.99_{\pm 0.21}$	$2.02_{\pm 0.04}$	$2.53_{\pm 0.40}$	$\mathbf{1.02}_{\pm 0.25}$
	U	$0.23_{\pm 0.06}$	$0.45_{\pm 0.12}$	$0.32_{\pm 0.05}$	$0.23_{\pm 0.13}$	$0.37_{\pm 0.11}$	$\mathbf{0.21}_{\pm 0.08}$
MO-Hopper-v2	HV($\times 10^5$)	$4.20_{\pm 0.39}$	$2.18_{\pm 0.03}$	$3.41_{\pm 0.27}$	$3.02_{\pm 0.35}$	$1.58_{\pm 0.17}$	$\mathbf{5.26}_{\pm 0.21}$
	SP($\times 10^1$)	$0.42_{\pm 0.02}$	$2.19_{\pm 0.34}$	$0.98_{\pm 0.16}$	$0.51_{\pm 0.46}$	$2.10_{\pm 0.15}$	$\mathbf{0.22}_{\pm 0.36}$
	U	$0.48_{\pm 0.13}$	$0.33_{\pm 0.12}$	$0.23_{\pm 0.12}$	$0.32_{\pm 0.16}$	$0.44_{\pm 0.21}$	$\mathbf{0.17}_{\pm 0.13}$
MO-Hopper-v3	HV($\times 10^7$)	$5.37_{\pm 0.56}$	$4.30_{\pm 0.14}$	$3.80_{\pm 0.17}$	$1.42_{\pm 0.22}$	$3.41_{\pm 0.38}$	$\mathbf{7.36}_{\pm 0.53}$
	SP($\times 10^1$)	$2.71_{\pm 0.08}$	$2.10_{\pm 0.32}$	$\mathbf{1.65}_{\pm 0.21}$	$3.02_{\pm 0.18}$	$2.91_{\pm 0.35}$	$2.50_{\pm 0.41}$
	U	$0.21_{\pm 0.19}$	$0.35_{\pm 0.10}$	$0.26_{\pm 0.15}$	$0.32_{\pm 0.27}$	$0.28_{\pm 0.19}$	$\mathbf{0.15}_{\pm 0.11}$
Average Rank	HV / SP / U	2.0 / 2.0 / 3.5	4.2 / 3.8 / 4.6	4.2 / 4.6 / 3.2	4.6 / 3.8 / 3.5	5.0 / 5.0 / 5.2	**1.0 / 1.8 / 1.0**

Table 1: Performance comparison on the MO-MuJoCo benchmark. The best performance is in bold.

Problem	Metric	Envelope	PD-MORL	RMORL	DROPS
FTN (d=5)	HV	$4.32_{\pm 0.73}$	$4.59_{\pm 0.21}$	$4.67_{\pm 0.95}$	$\mathbf{5.97}_{\pm 0.38}$
	SP	N/A	N/A	$\mathbf{0.09}_{\pm 0.01}$	$0.12_{\pm 0.05}$
	U	$0.42_{\pm 0.09}$	$0.43_{\pm 0.12}$	$0.37_{\pm 0.07}$	$\mathbf{0.29}_{\pm 0.10}$
FTN (d=6)	HV	$5.63_{\pm 0.44}$	$5.82_{\pm 0.88}$	$7.03_{\pm 0.31}$	$\mathbf{7.29}_{\pm 0.67}$
	SP	N/A	N/A	$0.16_{\pm 0.04}$	$\mathbf{0.16}_{\pm 0.02}$
	U	$0.45_{\pm 0.09}$	$0.50_{\pm 0.09}$	$0.41_{\pm 0.14}$	$\mathbf{0.32}_{\pm 0.08}$
FTN (d=7)	HV	$8.09_{\pm 1.07}$	$8.11_{\pm 0.52}$	$8.84_{\pm 0.79}$	$\mathbf{9.52}_{\pm 0.41}$
	SP	N/A	N/A	$0.15_{\pm 0.03}$	$\mathbf{0.13}_{\pm 0.02}$
	U	$0.63_{\pm 0.10}$	$0.59_{\pm 0.13}$	$0.47_{\pm 0.09}$	$\mathbf{0.25}_{\pm 0.06}$

Table 2: Performance comparison on the FTN benchmark. The HV value is scaled by 10^3 for readability, and N/A for SP indicates that the method finds only one solution.

(i) Quality Evaluation. Overall, DROPS performs best across the two benchmarks, achieving the top average rank on both HV and SP, where higher HV and lower SP indicate a higher quality of Pareto Set approximation. In terms of HV, DROPS ranks first on HV across all five MO-MuJoCo tasks and all FTN depths, with clear margins over strong baselines. In particular, DROPS surpasses the suboptimal model with ratios of 11.0% on MO-Walker2d-v2 and 7.7% on FTN (d=7). In terms of SP, DROPS produces competitively well-spread solutions on MO-MuJoCo. DROPS consistently returns multi-solution Pareto Set with competitive spacing on FTN, whereas some baselines degenerate to a single solution. These results suggest that DROPS improves Pareto Set quality and maintains diversity across both continuous and discrete benchmarks. ***(ii) Robustness Evaluation.*** As illustrated in Table 1 and Table 2, DROPS demonstrates superior robustness across all benchmark problems. Specifically, the performance advantage of DROPS over the second-best algorithm varies across different environments, achieving 27.27% on MO-Swimmer-v2 compared to PSL-MORL, 28.57% on MO-Hopper-v3 in comparison to RMORL, 21.61% on FTN (d=5) relative to RMORL, and 46.80% on FTN (d=7) when contrasted with RMORL, respectively. This improvement arises from combining hierarchical adversarial perturbations with contrastive drift alignment, in which we tighten the degradation bound under uncertainty and align drifted preferences in the trajectory-embedding space. The inclusion of $\mathcal{L}_{adv}$ promotes robustness of the policy network, while $\mathcal{L}_{cl}$ and $\mathcal{L}_{ds}$ guide the hypernetwork to converge to more diverse policies.

What is the impact of each component in DROPS, and which is most critical? The experimental results presented above demonstrate the superiority of our DROPS approach. While there are advantages compared to competing methods, the contributions of the main components still require clarification. We perform ablation experiments under consistent settings and training scheme, in which the specific ablations include: i. removal of adversarial loss (w/o $\mathcal{L}_{adv}$); ii. removal of contrastive loss (w/o $\mathcal{L}_{cl}$); and iii. removal of diversity loss (w/o $\mathcal{L}_{ds}$).

As reported in Table 3, the complete DROPS variant attains the best overall performance, achieving the highest HV while yielding the lowest SP and U. Removing either $\mathcal{L}_{adv}$ or $\mathcal{L}_{cl}$ causes a pronounced performance degradation, particularly in robustness, whereas deleting one of them provides limited improvement in Pareto Set quality. Moreover, excluding $\mathcal{L}_{ds}$ substantially impairs the

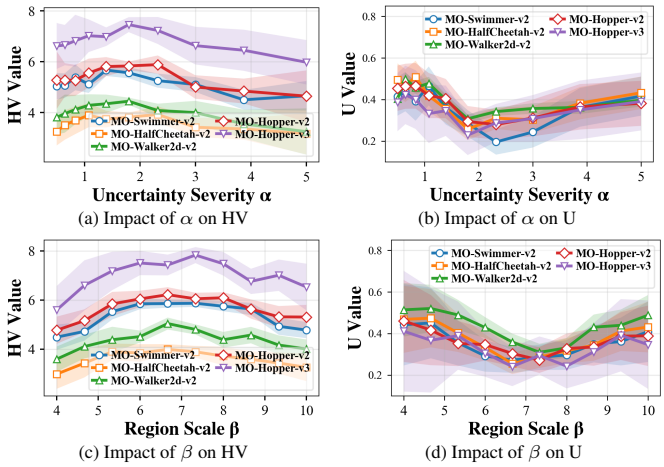
distribution of the learned Pareto Set, indicating that $\mathcal{L}_{ds}$ is essential for alleviating diversity loss and preventing policy collapse. Notably, $\mathcal{L}_{ds}$ also contributes to quality and robustness, likely by encouraging the hypernetwork to explore a broader range of latent directions; in conjunction with $\mathcal{L}_{adv}$ and $\mathcal{L}_{cl}$, it leads to a more accurate and robust Pareto Set approximation. These findings suggest that each component is critical to achieving the overall optimality and robustness of DROPS.

Problem	MO-Swimmer-v2			MO-HalfCheetah-v2			MO-Walker2d-v2			MO-Hopper-v2			MO-Hopper-v3		
Metrics	HV	SP	U	HV	SP	U	HV	SP	U	HV	SP	U	HV	SP	U
w/o $\mathcal{L}_{adv}$	2.53	0.88	0.49	1.82	1.29	0.34	3.58	1.48	0.33	4.11	0.92	0.23	5.03	2.68	0.48
w/o $\mathcal{L}_{cl}$	2.77	0.82	0.52	1.55	1.61	0.58	3.33	1.65	0.44	3.62	0.75	0.41	4.75	2.87	0.43
w/o $\mathcal{L}_{ds}$	4.45	1.03	0.21	2.00	2.52	0.26	3.97	2.06	0.27	4.90	1.26	0.32	6.88	2.91	0.23
DROPS	**4.97**	**0.77**	**0.16**	**2.45**	**0.90**	**0.18**	**4.05**	**1.02**	**0.21**	**5.26**	**0.22**	**0.17**	**7.36**	**2.50**	**0.15**

Table 3: Ablation study on the MO-MuJoCo benchmark. All the metric values are scaled to be consistent with those in Table 1.

How do hyperparameters affect the quality and robustness of the Pareto Set produced by DROPS? Our sensitivity analysis examines two key hyperparameters in DROPS: (i) the uncertainty severity for adversarial perturbations and (ii) the trust-region scale for contrastive learning. Specifically, we define $\alpha = |\zeta|/|\eta|$ to measure relative severity, and $\beta = |\delta|/|\epsilon|$ to quantify the region scale. These two parameters jointly determine the robustness of hypernetwork-based generation and the tolerance to preference drift.

As shown in Figure 3, a consistent pattern emerges across all MO-MuJoCo tasks: selecting a moderately small α together with a relatively large β yields higher HV and lower U, whereas either overly small or overly large values degrade performance. With a smaller α, uncertainties perturb the in-region and out-of-region preference more symmetrically, thereby improving robustness. In contrast, an excessively large β over-separates the trust region, leading to suboptimal trade-offs and less stable adaptation. Notably, the variance is minimized in the vicinity of the best-performing α and β, indicating that an appropriate severity and scale not only improve the quality of the Pareto Set, but also enhance robustness against preference drift.



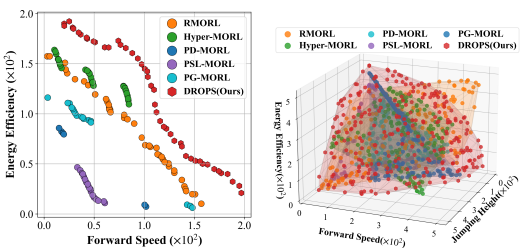
(a) Impact of α on HV (b) Impact of α on U

(c) Impact of β on HV (d) Impact of β on U

Figure 3: Sensitivity of DROPS to uncertainty severity and region scale on MO-MuJoCo tasks.

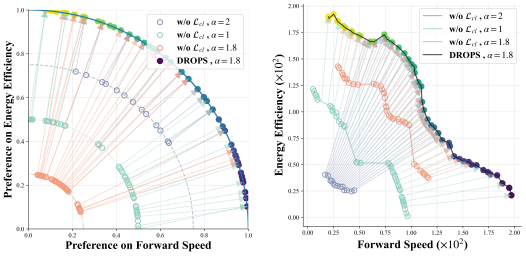
Can the generated Pareto Set provide an intuitive visualization of the performance? To facilitate a more intuitive comparison among methods, Figure 4 visualizes the Pareto Sets learned by each approach on a two-objective task, MO-HalfCheetah-v2, and a three-objective task, MO-Hopper-v3. As shown in Figure 4a, RMORL produces a Pareto Set that largely tracks the one obtained by DROPS; however, DROPS achieves a higher HV, indicating a better approximation quality. Although PD-MORL reports a smaller SP value on MO-Hopper-v3, Figure 4b reveals that its solutions remain clearly separated from those of DROPS. Overall, these results suggest that improving spacing alone does not necessarily translate into a more accurate Pareto Set approximation, whereas DROPS consistently exhibits both higher quality and stronger robustness under uncertainties.

Does DROPS robustly adapt to preference drift under uncertainties? To better visualize DROPS's adaptation behavior, Figure 5 compares DROPS with its ablation w/o $\mathcal{L}_{cl}$ on MO-HalfCheetah-v2 under different α. As shown in Figure 5a, DROPS yields a uniformly covered Pareto Set over the preference space, whereas removing $\mathcal{L}_{cl}$ leaves some preferences without feasible solutions. We attribute these gaps to uncertainty-induced preference drift, which causes certain



(a) MO-HalfCheetah-v2 (b) MO-Hopper-v3

Figure 4: Visualization of Pareto Sets obtained on different MO-MuJoCo tasks.



(a) Preference space (b) Objective space

Figure 5: Visualization of preference drift adaptation under uncertainties.

drifted preferences to fall outside the effective coverage of the learned generator. Consistently, in the objective space, as shown in Figure 5b, DROPS maintains a well-covered and competitive Pareto Set, while the ablated variant exhibits systematic degradation and partial collapse over large regions of the front. Overall, these results suggest that DROPS robustly adapts to preference drift.

5 Related Works

Pareto Set Learning. Recent PSL methods generate personalized policies for diverse user preferences by leveraging hypernetworks, ensuring comprehensive coverage of all trade-off possibilities. Hyper-MORL (Shu et al., 2024) learns a Pareto set in a high-dimensional policy parameter space using a single hypernetwork. PSL-MORL (Liu et al., 2025) introduces a decomposition-based framework where a hypernetwork generates parameters for each preference. DA-PSL (Lu et al., 2024), PSLGP (Xu et al., 2025), and CoPSL (Shang et al., 2024) explore various other variants of hypernetworks to address specific practical issues. However, these methods primarily optimize nominal Pareto coverage and do not explicitly account for uncertainty-induced changes in the preference-to-performance correspondence.

Uncertainties in MORL. Several studies improve robustness by explicitly modeling or adversarially perturbing uncertainty during policy learning. A2P (Liu et al., 2024), ARA-RL(Yang et al., 2024a), SortRL(Nie et al., 2024), and Rat (Bai et al., 2025) extend the forms of adversarial perturbations to keep the training stable while effectively improving robustness. RMORL (He et al., 2024) employs a zero-sum game in the adversarial agent and devises an adversarial defense mechanism. GRAD (Liang et al., 2024), PAL (Rajeswaran et al., 2020), and LRP (Korkmaz, 2023) reformulated the specific problem within a game-theoretic framework, subsequently employing concepts from game theory to achieve stable control. Although these methods demonstrate a certain level of robustness, their adversarial perturbation strategies are characterized as disturbances within a predefined bounded distribution. However, the forms of uncertainties encountered in natural environments extend beyond merely disturbances.

6 Conclusion

This paper proposes DROPS, which builds on the PSL framework for learning a robust Pareto Set against observational and environmental uncertainties. DROPS identifies ***preference drift*** caused by uncertainties as an underexplored factor of robustness failure. By integrating hierarchical adversarial perturbations and contrastive drift alignment, DROPS reduces uncertainty-induced return displacement while explicitly regularizing the preference-to-policy mapping to preserve the intended preference-to-performance correspondence. Experimental results show that DROPS achieves the best overall performance compared to competing methods on benchmark problems, demonstrating its superior robustness and adaptability to uncertainties. Our future work will investigate preference drift under more general partial observability, and extend DROPS to higher-dimensional multi-objective and real-world robotic control tasks. We also plan to develop stronger theoretical guarantees for preserving preference-to-policy correspondence under uncertainty.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language editing, improving clarity and readability, and refining the presentation of the manuscript. We did not use generative AI tools to generate experimental data or to independently conduct experiments or draw scientific conclusions, and other required disclosure tasks are not applicable to this work. All AI-assisted content was carefully reviewed, revised, and verified by the authors to ensure its technical accuracy and consistency with the intended scientific claims. We take full responsibility for the final content of this work, including any text or artifacts produced with the aid of generative AI.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pp. 22–31. PMLR, 2017.

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

Fengshuo Bai, Runze Liu, Yali Du, Ying Wen, and Yaodong Yang. Rat: Adversarial attacks on deep reinforcement agents for targeted behaviors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 15453–15461, 2025.

Toygun Basaklar, Suat Gumussoy, and Umit Ogras. Pd-morl: Preference-driven multi-objective reinforcement learning algorithm. In *The Eleventh International Conference on Learning Representations*, 2022.

Lauro Langosco Di Langosco, Jack Koch, Lee D Sharkey, Jacob Pfau, and David Krueger. Goal misgeneralization in deep reinforcement learning. In *International Conference on Machine Learning*, pp. 12004–12019. PMLR, 2022.

Kamyar Ghasemipour, Shixiang Shane Gu, and Ofir Nachum. Why so pessimistic? estimating uncertainties for offline rl through ensembles, and why their independence matters. *Advances in Neural Information Processing Systems*, 35:18267–18281, 2022.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 297–304. JMLR Workshop and Conference Proceedings, 2010.

Xiangkun He, Jianye Hao, Xu Chen, Jun Wang, Xuewu Ji, and Chen Lv. Robust multiobjective reinforcement learning considering environmental uncertainties. *IEEE Transactions on Neural Networks and Learning Systems*, 36(4):6368–6382, 2024.

Siwei Jiang, Yew-Soon Ong, Jie Zhang, and Liang Feng. Consistencies and contradictions of performance metrics in multiobjective optimization. *IEEE transactions on cybernetics*, 44(12):2391–2404, 2014.

Ezgi Korkmaz. Adversarial robust deep reinforcement learning requires redefining robustness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 8369–8377, 2023.

Yongyuan Liang, Yanchao Sun, Ruijie Zheng, Xiangyu Liu, Benjamin Eysenbach, Tuomas Sandholm, Furong Huang, and Stephen McAleer. Game-theoretic robust rl handles temporally-coupled perturbations. In *12th International Conference on Learning Representations, ICLR 2024*, 2024.

Erlong Liu, Yu-Chang Wu, Xiaobin Huang, Chengrui Gao, Ren-Jian Wang, Ke Xue, and Chao Qian. Pareto set learning for multi-objective reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 18789–18797, 2025.

Qianmei Liu, Yufei Kuang, and Jie Wang. Robust deep reinforcement learning with adaptive adversarial perturbations in action space. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2024.

Owen Lockwood and Mei Si. A review of uncertainty for deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 18, pp. 155–162, 2022.

Yongfan Lu, Bingdong Li, and Aimin Zhou. Are you concerned about limited function evaluations: data-augmented pareto set learning for expensive multi-objective optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 14202–14210, 2024.

Kanghua Mo, Peigen Ye, Xiaojun Ren, Shaowei Wang, Wenjun Li, and Jin Li. Security and privacy issues in deep reinforcement learning: Threats and countermeasures. *ACM Computing Surveys*, 56(6):1–39, 2024.

Buqing Nie, Jingtian Ji, Yangqing Fu, and Yue Gao. Improve robustness of reinforcement learning against observation perturbations via l lipschitz policy networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 14457–14465, 2024.

Aravind Rajeswaran, Igor Mordatch, and Vikash Kumar. A game theoretic framework for model based reinforcement learning. In *International conference on machine learning*, pp. 7953–7963. PMLR, 2020.

Chikai Shang, Rongguang Ye, Jiaqi Jiang, and Fangqing Gu. Collaborative pareto set learning in multiple multi-objective optimization problems. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2024.

Ke Shang, Hisao Ishibuchi, Linjun He, and Lie Meng Pang. A survey on the hypervolume indicator in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 25(1):1–20, 2020.

Tianye Shu, Ke Shang, Cheng Gong, Yang Nan, and Hisao Ishibuchi. Learning pareto set for multi-objective continuous robot control. In *33rd International Joint Conference on Artificial Intelligence (IJCAI 2024)*, pp. 4920–4928. International Joint Conferences on Artificial Intelligence, 2024.

Siddharth Singi, Zhanpeng He, Alvin Pan, Sandip Patel, Gunnar A Sigurdsson, Robinson Piramuthu, Shuran Song, and Matei Ciocarlie. Decision making for human-in-the-loop robotic agents via uncertainty-aware reinforcement learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7939–7945. IEEE, 2024.

Chen Tang, Ben Abbatematteo, Jiaheng Hu, Rohan Chandra, Roberto Martín-Martín, and Peter Stone. Deep reinforcement learning for robotics: A survey of real-world successes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 28694–28698, 2025.

Ran Wei, Siliang Zeng, Chenliang Li, Alfredo Garcia, Anthony D McDonald, and Mingyi Hong. A bayesian approach to robust inverse reinforcement learning. In *Conference on Robot Learning*, pp. 2304–2322. PMLR, 2023.

Jie Xu, Yunsheng Tian, Pingchuan Ma, Daniela Rus, Shinjiro Sueda, and Wojciech Matusik. Prediction-guided multi-objective reinforcement learning for continuous robot control. In *International conference on machine learning*, pp. 10607–10616. PMLR, 2020.

Meng Xu, Yi Mei, Fangfang Zhang, Yew Soon Ong, and Mengjie Zhang. Pareto set learning through genetic programming for multi-objective dynamic scheduling. *IEEE Transactions on Evolutionary Computation*, 2025.

Jiacheng Yang, Yuanda Wang, Lu Dong, Lei Xue, and Changyin Sun. Active robust adversarial reinforcement learning under temporally-coupled perturbations. *IEEE Transactions on Artificial Intelligence*, 2024a.

Rui Yang, Jie Wang, Guoping Wu, and Bin Li. Uncertainty-based offline variational bayesian reinforcement learning for robustness under diverse data corruptions. *Advances in Neural Information Processing Systems*, 37:39748–39783, 2024b.

Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. *Advances in neural information processing systems*, 32, 2019.

A BENCHMARK DETAILS

A.1 MO-MUJOCO

To evaluate the quality and robustness of the Pareto Set learned by DROPS, we conduct experiments on five problems from the multi-objective robotic control benchmark suite MO-MuJoCo (Basaklar et al., 2022), as summarized in Table 4. MO-MuJoCo is a widely used MORL benchmark built upon the MuJoCo physics engine. Different from the original clean benchmark, we introduce both step-level interaction uncertainty and episode-level environmental variation to evaluate whether preference-conditioned policies can preserve their intended trade-offs under uncertainty. These uncertainty settings are consistent with the uncertainty formulation introduced in the main paper.

Problem	m	State Space	Action Space
MO-Swimmer-v2	2	$\mathcal{S} \in \mathbb{R}^8$	$\mathcal{A} \in \mathbb{R}^2$
MO-HalfCheetah-v2	2	$\mathcal{S} \in \mathbb{R}^{17}$	$\mathcal{A} \in \mathbb{R}^6$
MO-Walker2d-v2	2	$\mathcal{S} \in \mathbb{R}^{17}$	$\mathcal{A} \in \mathbb{R}^6$
MO-Hopper-v2	2	$\mathcal{S} \in \mathbb{R}^{11}$	$\mathcal{A} \in \mathbb{R}^3$
MO-Hopper-v3	3	$\mathcal{S} \in \mathbb{R}^{11}$	$\mathcal{A} \in \mathbb{R}^3$

Table 4: Dimension information of the five MO-MuJoCo test problems.

- **MO-Swimmer-v2**: The agent is a simplified three-link swimming robot operating in a viscous fluid environment, with $\mathcal{S} \subseteq \mathbb{R}^8$ and $\mathcal{A} \subseteq \mathbb{R}^2$. The task considers the trade-off between forward motion and energy consumption, where the preference vector ω specifies the desired compromise between the objectives.

- **MO-HalfCheetah-v2**: The agent is a planar robotic system resembling a cheetah, with $\mathcal{S} \subseteq \mathbb{R}^{17}$ and $\mathcal{A} \subseteq \mathbb{R}^6$. The task optimizes forward motion and energy efficiency. A preference vector ω specifies the intended trade-off, from which the hypernetwork generates the corresponding preference-conditioned policy.

- **MO-Walker2d-v2**: The agent is a planar bipedal robot with $\mathcal{S} \subseteq \mathbb{R}^{17}$ and $\mathcal{A} \subseteq \mathbb{R}^6$. Its objectives characterize competing aspects of locomotion performance, and the preference vector ω determines the desired trade-off realized by the generated policy.

- **MO-Hopper-v2**: The agent is a planar one-legged robot with $\mathcal{S} \subseteq \mathbb{R}^{11}$ and $\mathcal{A} \subseteq \mathbb{R}^3$. The policy controls the joint torques to realize different locomotion trade-offs according to the input preference ω.

- **MO-Hopper-v3**: The agent is a planar one-legged robot with $\mathcal{S} \subseteq \mathbb{R}^{11}$ and $\mathcal{A} \subseteq \mathbb{R}^3$. In contrast to the two-objective tasks above, MO-Hopper-v3 contains three objectives involving jumping height, forward velocity, and energy consumption. The preference vector ω therefore specifies a trade-off in a three-dimensional objective space.

A.2 FRUIT TREE NAVIGATION

The Fruit Tree Navigation (FTN) environment (Yang et al., 2019) is a discrete multi-objective reinforcement learning benchmark in which the agent navigates through a full binary tree of depth $d \in \{5, 6, 7\}$. At each non-terminal node, the agent selects one of two actions and continues along the corresponding branch until reaching a leaf node. The reward vectors are constructed such that the leaves form a convex coverage set: for each Pareto-optimal leaf, there exists at least one preference vector ω under which the corresponding reward vector is optimal. Given ω, the agent seeks a path maximizing the scalarized utility $\omega^\top \mathbf{r}$.

FTN therefore provides a six-objective test bed that complements the two- and three-objective continuous-control tasks in MO-MuJoCo. This setting is particularly useful for evaluating whether a learned preference-to-policy mapping remains reliable as the dimensionality of the preference and objective spaces increases.

Parameter	Uncertainty Injection	Temporal Level	Channel
Gravity	$g' = g_0(1+\epsilon),\ \epsilon \sim \mathcal{U}(-0.3\vartheta, 0.3\vartheta)$	Episode-level	Environment
Body mass	$m' = m(1+\epsilon),\ \epsilon \sim \mathcal{U}(-0.5\vartheta, 0.5\vartheta)$	Episode-level	Environment
Joint damping	$d' = d(1+\epsilon),\ \epsilon \sim \mathcal{U}(-1.0\vartheta, 1.0\vartheta)$	Episode-level	Environment
Contact friction	$\mu' = \mu(1+\epsilon),\ \epsilon \sim \mathcal{U}(-0.5\vartheta, 0.5\vartheta)$	Episode-level	Environment
Observation disturb	$o' = o + \eta, \quad \eta \sim \mathcal{N}\big(0, (\vartheta\sigma_o)^2\big)$	Step-level	Observation
Observation mask	$o' = o \odot m, \quad m \sim \text{Bernoulli}(1-p),\ p = \vartheta p_{\max}$	Step-level	Observation
Action disturb	$a' = \text{clip}(a + \eta), \quad \eta \sim \mathcal{N}\big(0, (\vartheta\sigma_a)^2\big)$	Step-level	Action
Action delay	$a'_t = a_{t-d}, \quad d = \lceil \vartheta d_{\max} \rceil$	Step-level	Action

Table 5: Uncertainty injection for MO-MuJoCo. The unified intensity scalar $\vartheta \in [0, 1]$ controls the magnitude of both step-level interaction corruption and episode-level environment variation. Episode-level perturbations are sampled at `reset()` and remain fixed throughout an episode, whereas step-level perturbations are applied during interaction.

A.3 Benchmark Uncertainty

Consistent with the main-text formulation, we distinguish uncertainty according to its temporal persistence. A step-level uncertainty realization acts during individual agent–environment interactions, whereas an episode-level realization is sampled once at the beginning of an episode and remains fixed throughout that episode.

We denote a generic uncertainty realization by $u = (\nu, \xi)$, where ν represents step-level interaction corruption and ξ represents an episode-level environment configuration. The distinction between step-level and episode-level uncertainty refers to temporal persistence rather than to a single physical channel. Accordingly, uncertainty may affect observations, executed actions, environment transitions, or task parameters. To avoid confusion with the preference-space perturbation δ used in the main text to construct $\omega_o = \omega + \delta$, we denote the benchmark-level uncertainty intensity by $\vartheta \in [0, 1]$. Here, $\vartheta = 0$ corresponds to the nominal benchmark, whereas $\vartheta = 1$ reaches the maximum predefined perturbation range.

MO-MuJoCo. Table 5 summarizes the uncertainty realizations considered for MO-MuJoCo. Episode-level uncertainty modifies the physical environment parameters at `reset()` and keeps the resulting configuration fixed within the episode. Step-level uncertainty acts during interaction and changes either the observation received by the policy or the executed action.

For ***episode-level*** variation, the simulator parameters are perturbed using multiplicative shifts of the general form $\theta' = \theta(1+\epsilon)$. These perturbations model persistent changes in physical properties such as gravity, inertial properties, damping, and contact conditions. They therefore instantiate the environment configuration ξ in the main-text formulation. For ***step-level*** uncertainty, observation disturbance adds Gaussian noise to the observation, whereas observation masking randomly suppresses observation dimensions. These operators instantiate the observation corruption component of ν. Action disturbance and action delay alter the executed action after it has been generated by the preference-conditioned policy. From the viewpoint of the degradation analysis, such action corruption can be absorbed into the effective perturbed action distribution $\Pi_{\omega,u}$.

Importantly, these perturbations are applied to preference-conditioned policies rather than to separate "positive" or "negative" preference classes. This is consistent with the current formulation of DROPS, in which preference drift is characterized through the uncertainty-induced change of the preference-to-return correspondence.

FTN. We use the same temporal hierarchy for FTN. Table 6 summarizes the corresponding uncertainty operators. For ***step-level*** uncertainty, an action flip reverses the selected action with probability $p = \vartheta p_{\max}$, while branch slip sends the agent to an unintended child node with the same probability. Observation corruption is introduced through additive Gaussian noise or random masking. These operators create uncertainty in the interaction process while leaving the underlying preference vector unchanged. For ***episode-level*** uncertainty, the task instance is changed once at `reset()` and remains fixed within the episode. Subtree swapping modifies the transition structure. Leaf reward

Parameter	Uncertainty Injection	Temporal Level	Channel
Leaf reward noise	$r'_{\text{leaf}} = r_{\text{leaf}} + \eta, \quad \eta \sim \mathcal{N}\big(0, (\vartheta\sigma_r)^2\big)$	Episode-level	Task / reward
Reward scaling drift	$r' = s \odot r, \quad s_i \sim \mathcal{U}(1 - \vartheta r_s, 1 + \vartheta r_s)$	Episode-level	Task / reward
Subtree swap	swap $m = \lceil \vartheta m_{\max} \rceil$ internal nodes at `reset()`	Episode-level	Environment
Action flip	$a' = \text{flip}(a)$ w.p. $p = \vartheta p_{\max}$	Step-level	Action
Branch slip	transition to an unintended child w.p. $p = \vartheta p_{\max}$	Step-level	Transition
Observation disturb	$o' = o + \eta, \quad \eta \sim \mathcal{N}\big(0, (\vartheta\sigma_o)^2\big)$	Step-level	Observation
Observation mask	$o' = o \odot m, \quad m \sim \text{Bernoulli}(1-p),\ p = \vartheta p_{\max}$	Step-level	Observation

Table 6: Uncertainty injection for FTN. The same benchmark-level uncertainty intensity $\vartheta \in [0, 1]$ is used as in MO-MuJoCo. Episode-level uncertainty changes the task instance at `reset()`, whereas step-level uncertainty acts during individual decision steps.

Hyperparameter	MO-Swimmer-v2	MO-Walker2d-v2	MO-HalfCheetah-v2	MO-Hopper-v2	MO-Hopper-v3
Total steps	10^6	10^6	10^6	10^6	10^6
Minibatch size	512	512	512	512	512
Discount factor γ	0.995	0.995	0.995	0.995	0.995
Soft update τ	0.003	0.003	0.003	0.003	0.003
Replay buffer size	3×10^6	3×10^6	3×10^6	3×10^6	3×10^6
Critic LR	2×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-4}
Critic hidden layers / neurons	1 / 400	1 / 400	1 / 400	1 / 400	1 / 400
Actor LR	2×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-4}	2×10^{-4}
Actor hidden layers / neurons	1 / 400	1 / 400	1 / 400	1 / 400	1 / 400
Policy update delay	10	10	10	10	**20**
Parameter fusion coefficient	0.05	0.05	0.05	0.05	0.05
Exploration noise std.	0.20	0.20	0.20	**0.25**	**0.25**
Noise clipping limit	0.7	0.7	0.7	0.7	0.7
Loss coefficient	15	15	15	15	15

Table 7: Hyperparameter settings of DROPS on MO-MuJoCo tasks under perturbed environments.

noise and reward scaling alter the objective values associated with the terminal outcomes and therefore directly modify the uncertain return map F_u.

The formal definition of preference drift is unaffected by the physical source of uncertainty because it depends only on the correspondence between the nominal return map F_0 and the uncertain return map F_u. When reward-shift variants are considered, their effect is therefore directly reflected in $F_u(\omega)$ and in the resulting preference drift measure. The degradation bound in the main text specifically characterizes the observation- and transition-induced components under bounded rewards.

B DROPS Training Details

Table 7 reports the hyperparameter configuration used to train DROPS on the perturbed MO-MuJoCo tasks.

Compared with clean off-policy continuous-control training, environment variation and interaction corruption introduce stronger heterogeneity and distributional shift into the replay data. We therefore adopt a conservative training configuration to improve optimization stability while learning preference-conditioned policies under uncertainty.

Specifically, we use learning rates of 2×10^{-4} for both the actor and critic and a soft target-update coefficient of 0.003. The replay buffer contains 3×10^6 transitions and the minibatch size is set to 512. These settings expose the learner to a broader range of perturbed experience while reducing gradient variance. Following delayed policy updates in deterministic policy-gradient methods, the policy update delay is set to 10 for most tasks and to 20 for MO-Hopper-v3. The exploration-noise standard deviation is increased to 0.25 for the two Hopper tasks and is clipped at 0.7 to avoid excessively aggressive actions. Unless otherwise specified in Table 7, the remaining settings are shared across tasks. Overall, these support the two complementary objectives of DROPS: maintaining Pareto Set quality while preserving the preference-to-policy correspondence under uncertainty.

B.1 Preference Sampling and Trust-Region Construction

We next provide implementation details for the drift-aware trust region introduced in the main paper.

The preference space is the simplex $\Delta^{m-1} = \{\omega \in \mathbb{R}_+^m \mid \mathbf{1}^\top \omega = 1\}$. For each training instance, an anchor preference $\omega \in \Delta^{m-1}$ is first sampled from the preference distribution $q(\omega)$. The hypernetwork generates the preference-conditioned policy π_ω. Given another preference ω', its behavioral proximity to ω is characterized through $\bar{D}(\pi_{\omega'} \| \pi_\omega)$, which is the KL divergence between the two policy distributions averaged over the reference states used during training.

The preference-specific trust region is $T_\omega(\rho) = \{\omega' \in \Delta^{m-1} \mid \bar{D}(\pi_{\omega'} \| \pi_\omega) \le \rho\}$. Following the main-text construction, an in-region preference is generated as $\omega_i = \omega + \epsilon$, with the resulting preference constrained to Δ^{m-1}. Under the local Lipschitz condition introduced in the main text, $\|\epsilon\|_2 \le \rho/L$ provides a sufficient condition for generating an in-region candidate. For the out-of-region sample, we first generate $\widetilde{\omega}_o = \omega + \delta$, $\|\delta\|_2 > \|\epsilon\|_2$, and retain it only when $\bar{D}(\pi_{\widetilde{\omega}_o} \| \pi_\omega) > \rho$. The accepted sample is denoted by ω_o. This explicit membership test is important because the one-sided Lipschitz upper bound does not imply that a preference with a sufficiently large Euclidean distance must lie outside the policy trust region. Accordingly, ω_i is used as a behaviorally local reference, whereas ω_o is treated as a hard drift candidate.

Neither ω_i nor ω_o changes the formal definition of preference drift. Actual preference drift is defined through the uncertainty-induced change of the preference-to-return correspondence, whereas the trust region provides tractable training samples for learning that correspondence robustly.

B.2 Preference Drift Evaluation

The main paper defines preference drift using the set-valued inverse correspondence

$$\Omega_u^\star(\omega) = \arg\min_{\omega' \in \Delta^{m-1}} \|F_0(\omega') - F_u(\omega)\|_2 , \tag{8}$$

where $F_0(\omega)$ denotes the nominal return and $F_u(\omega)$ denotes the return realized under uncertainty u.

The corresponding drift magnitude is

$$d_{\mathrm{PD}}(\omega; u) = \mathrm{dist}(\omega, \Omega_u^\star(\omega)) = \min_{\bar{\omega} \in \Omega_u^\star(\omega)} \|\omega - \bar{\omega}\|_2 . \tag{9}$$

In experiments, the continuous preference space is approximated by a finite evaluation set

$$\mathcal{W} = \{\omega_1, \ldots, \omega_N\} \subset \Delta^{m-1}.$$

For each $\omega_i \in \mathcal{W}$, we first estimate its nominal return $\widehat{F}_0(\omega_i)$ and its uncertain return $\widehat{F}_u(\omega_i)$ from rollouts. The empirical inverse correspondence is then computed as

$$\widehat{\Omega}_u^\star(\omega_i) = \arg\min_{\omega_j \in \mathcal{W}} \left\| \widehat{F}_0(\omega_j) - \widehat{F}_u(\omega_i) \right\|_2 . \tag{10}$$

The empirical preference drift of ω_i is

$$\widehat{d}_{\mathrm{PD}}(\omega_i; u) = \min_{\omega_j \in \widehat{\Omega}_u^\star(\omega_i)} \|\omega_i - \omega_j\|_2 . \tag{11}$$

Finally, we report the preference drift degree by

$$\mathrm{U} = \frac{1}{N} \sum_{i=1}^{N} \widehat{d}_{\mathrm{PD}}(\omega_i; u). \tag{12}$$

A smaller U indicates that uncertain rollouts remain associated with nominal returns corresponding to nearby preferences and therefore preserve the intended preference-to-policy mapping more effectively. In contrast to HV and SP, which evaluate the geometry and quality of the Pareto Set, U explicitly measures the preference-space displacement introduced in Definition 1 of the main paper.

This metric also distinguishes the roles of the two major robustness components in DROPS. The adversarial loss $\mathcal{L}_{adv}$ limits uncertainty-induced return displacement, thereby reducing an important source of preference drift. The contrastive loss $\mathcal{L}_{cl}$ directly regularizes the preference-conditioned behavioral correspondence, while $\mathcal{L}_{ds}$ prevents this alignment from being achieved through a degenerate collapse of the generated policies.

Problem	Metrics	Hyper-MORL	PD-MORL	PSL-MORL	PG-MORL	**DROPS**
MO-HalfCheetah-v2	HV($\times 10^6$)	$5.53_{\pm 0.21}$	$5.89_{\pm 0.13}$	$5.78_{\pm 0.19}$	$5.75_{\pm 0.12}$	$\mathbf{5.92}_{\pm 0.10}$
	SP($\times 10^3$)	$0.29_{\pm 0.07}$	$0.47_{\pm 0.10}$	$0.18_{\pm 0.23}$	$0.44_{\pm 0.18}$	$\mathbf{0.12}_{\pm 0.09}$
MO-Walker2d-v2	HV($\times 10^6$)	$5.37_{\pm 0.38}$	$5.08_{\pm 0.42}$	$5.36_{\pm 0.25}$	$4.41_{\pm 0.22}$	$\mathbf{5.43}_{\pm 0.15}$
	SP($\times 10^4$)	$0.07_{\pm 0.29}$	$0.03_{\pm 0.08}$	$\mathbf{0.01}_{\pm 0.05}$	$0.04_{\pm 0.03}$	$0.19_{\pm 0.02}$
MO-Hopper-v3	HV($\times 10^{10}$)	$3.36_{\pm 0.26}$	$2.45_{\pm 0.11}$	$1.57_{\pm 0.29}$	$3.08_{\pm 0.31}$	$\mathbf{3.69}_{\pm 0.33}$
	SP($\times 10^7$)	$2.25_{\pm 0.30}$	$1.79_{\pm 0.27}$	$3.19_{\pm 0.20}$	$2.87_{\pm 0.33}$	$\mathbf{1.29}_{\pm 0.25}$

Table 8: Performance comparison between DROPS and competing methods on the non-perturbed MO-MuJoCo benchmark in terms of Hypervolume (HV) and Spacing (SP). Higher HV and lower SP indicate better performance. Mean values and standard deviations over eight runs are reported. The best results are highlighted in bold.

C Results on Non-Perturbed MO-MuJoCo

To verify that preference-drift robustness does not require sacrificing nominal Pareto Set quality, we additionally evaluate DROPS in the non-perturbed MO-MuJoCo environments. Table 8 reports the corresponding results in terms of Hypervolume (HV) and Spacing (SP).

Overall, DROPS remains competitive in the non-perturbed setting, indicating that the robustness mechanisms introduced for preference drift do not require sacrificing the quality of the nominal Pareto Set. On MO-HalfCheetah-v2, DROPS obtains both the highest HV and the lowest SP among the compared methods, although the performance differences in the clean environment remain relatively small. On MO-Walker2d-v2, DROPS achieves the highest HV, while PSL-MORL obtains a lower SP. This indicates that DROPS achieves strong nominal Pareto quality, whereas the competing method yields a more uniformly spaced approximation under this metric. On MO-Hopper-v3, DROPS obtains both the highest HV and the lowest SP. These clean-setting results complement the uncertainty experiments in the main paper. They indicate that DROPS preserves competitive nominal performance while its principal advantage emerges when uncertainty alters the realized preference-to-policy correspondence.