Distributionally Robust Domain Randomization with Learned Risk-Sensitive Dynamics Samplers

Sukchul Jeong
Seoul National University
tjrcjf410@snu.ac.kr

Insoon Yang*
Seoul National University
insoonyang@snu.ac.kr

Abstract

Domain randomization (DR) improves sim-to-real transfer by training policies over randomized simulator dynamics, but standard DR optimizes average performance under a fixed sampler and can miss rare failure-prone domains. We propose Risk-Sensitive Domain Randomization (RSDR), a distributionally robust formulation of episodic DR that evaluates each policy against the worst-case distribution over dynamics parameters under a fixed-temperature KL-regularized entropic-risk objective. The resulting soft adversary has an exponential-tilting form, yielding a single temperature-controlled sampler family: negative temperatures emphasize low-return domains for robustness, zero corresponds to uniform DR, and positive temperatures give an optimistic sampler closely related to curriculum-style DR. Because this target changes with the policy and returns are observed only through rollouts, RSDR learns an amortized dynamics sampler by reverse-KL variational inference while reusing on-policy PPO trajectories. Across six domain-randomized MuJoCo Playground tasks, robust RSDR improves CVaR and minimum-return metrics over uniform and adaptive DR baselines. It also outperforms best-tuned EPOpt on the four tasks evaluated with EPOpt.

1 Introduction

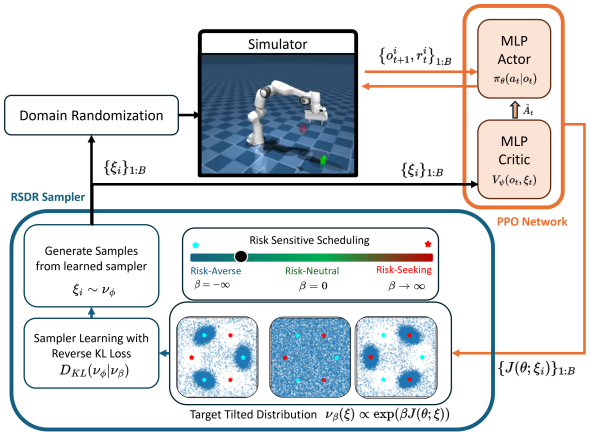
Sim-to-real transfer remains a central challenge in deploying reinforcement learning (RL) policies on physical systems. Real-world dynamics depend on quantities such as masses, friction, actuator gains, and contacts, which are difficult to model precisely and may drift over time. Domain randomization (DR) addresses this uncertainty by training in a simulator that randomly perturbs latent dynamics parameters, encouraging policies that generalize beyond a single nominal model [Tobin et al., 2017, Peng et al., 2018]. However, standard DR typically relies on a fixed sampling distribution (often uniform over a user-specified range), even though the true deployment distribution is unknown and rarely matches the training sampler.

A key mismatch is that failures in sim-to-real are governed by the *left tail* of performance over dynamics variations. Optimizing the expected return under a fixed sampler can produce policies that perform well on average while remaining brittle to adverse, low-probability dynamics realizations. This brittleness is exacerbated when the randomization range is wide or when dynamics exhibit non-stationarity at deployment. Motivated by this, we ask: *Can we learn a dynamics sampler that explicitly targets adverse dynamics in a principled way, while remaining compatible with standard deep RL training pipelines?*

Existing approaches partially address sampler design but do not fully resolve the robustness objective. Uniform DR (UDR) is simple but offers no mechanism to emphasize failure-prone regions. Curriculum and self-paced DR methods learn adaptive samplers that prioritize "easy" dynamics early

*Corresponding author

40th Conference on Neural Information Processing Systems (NeurIPS 2026).



Algorithm 1 Risk-Sensitive Domain Randomization

Input: θ_0, ϕ_0, β, batch size B, iterations T

for $t = 0$ **to** $T-1$ **do**
  Sample $\{\xi_i\}_{i=1}^{B} \sim \nu_{\phi_t}$
  Roll out π_{θ_t} in $\mathcal{M}_{\xi_i}$ to obtain $\{\tau_i\}_{i=1}^{B}$
  Estimate $\widehat{J}_i$ from τ_i via Eq. (1)
  Update ϕ_{t+1} by descending $\mathcal{L}_{RKL}(\phi; \theta_t, \beta)$ using Eq. (7)
  Update $\theta_{t+1} \leftarrow \mathrm{PPO}(\theta_t; \{\tau_i\}_{i=1}^{B})$
end for

Figure 1: Overview of our framework (left) and its training procedure (right). RSDR alternates between policy optimization under the current domain sampler and sampler optimization toward the exponentially tilted target distribution $\nu_\beta \propto \exp(\beta J_\pi)\nu_0$ induced by the current policy. The risk parameter β controls the degree of risk sensitivity in the sampler; in the robust regime emphasized in this paper, negative β increases sampling mass on adverse domains. In practice, we update the policy with PPO [Schulman et al., 2017] and reuse the same on-policy rollouts to update the sampler, so sampler learning incurs no additional environment interactions.

in training and gradually expand coverage [Akkaya et al., 2019, Tiboni et al., 2024, Curtis et al., 2025]; these methods are effective for exploration and learning progress, but they primarily bias toward the *right tail* of performance. At the other extreme, robust RL methods such as EPOpt directly optimize empirical CVaR objectives [Rajeswaran et al., 2016], but require substantially more rollouts to estimate tail quantiles and are often sensitive to hyperparameters. Classical distributionally robust MDP formulations typically model per-step uncertainty in transition kernels under rectangularity assumptions [Xu and Mannor, 2010, Wiesemann et al., 2013, Goyal and Grand-Clement, 2018], which do not naturally capture *episode-level* latent dynamics parameters that are sampled once per episode and induce correlated shifts across the trajectory.

We propose *Risk-Sensitive Domain Randomization* (RSDR), a KL-regularized distributionally robust formulation of episodic DR. Instead of committing to a single training sampler, we consider the worst-case dynamics-parameter distribution under a KL-regularized objective relative to a reference sampler $\nu_0(\xi)$ (e.g., uniform over the randomization range). This yields a closed-form *soft adversary* over dynamics parameters ξ: the worst-case distribution is an *exponentially tilted* reweighting by the return $J(\theta;\xi)$ of policy π_θ, i.e.,

$$\nu_\beta(\xi) \propto \exp\big(\beta\, J(\theta;\xi)\big)\, \nu_0(\xi),$$

where the temperature parameter β controls the degree of risk sensitivity. In the robust regime studied in this paper, negative β emphasizes low-return (failure-prone) dynamics, while $\beta = 0$ recovers the reference sampler; positive β emphasizes high-return dynamics and recovers an optimistic curriculum regime (See Figure 2).

Turning this principle into a practical algorithm is non-trivial because the target distribution depends on the current policy and returns are accessible only through rollouts. RSDR learns an expressive dynamics sampler via reverse-KL variational inference toward a policy-induced tilted target distribution, and alternates between sampler and policy updates while reusing the same on-policy trajectories (e.g., PPO rollouts), requiring no additional environment interactions for sampler learning. Beyond the robust regime studied here, the same tilted family also admits an optimistic interpretation; we discuss this secondary connection to curriculum-style DR.

Contributions.

- We formulate robustness in domain randomization at the level of episode-wise latent dynamics parameters, rather than rectangular state-action transition uncertainty. Starting from KL-DRO,

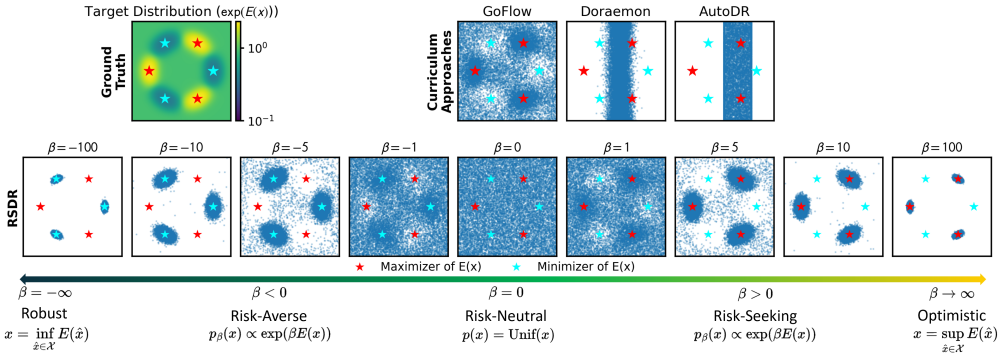


Figure 2: Illustration of the tilted-family view used by RSDR. The sampler targets $p_\beta(X) \propto \exp(\beta E(X))$, where $E(X)$ is defined in Appendix G.1. Curriculum-based DR methods can be viewed through the same tilted-family lens, corresponding most closely to the $\beta > 0$ risk-seeking regime (Appendix C). By controlling the risk parameter β, RSDR can sample from regions near maximizers (easy regions) or minimizers (hard regions).

we derive a fixed-temperature KL-regularized (entropic-risk) objective whose soft adversary is the exponentially tilted sampler specialized to episodic DR.

- We propose RSDR, an amortized sampler-learning algorithm that tracks the policy-dependent tilted target with reverse-KL variational inference and reuses on-policy rollouts for both PPO and sampler updates, avoiding a separate inner-loop adversarial sampler.
- We clarify the role of the temperature β: $\beta < 0$ gives the robust regime used in our experiments, $\beta = 0$ corresponds to UDR, and $\beta > 0$ provides an optimistic/curriculum-style counterpart.
- On six MuJoCo Playground domain-randomization benchmarks, RSDR improves left-tail robustness over UDR and adaptive DR baselines. On the four tasks evaluated with EPOpt, it also outperforms the best-tuned EPOpt configurations.

2 Related work

Domain randomization as learning under latent dynamics variation. Domain randomization (DR) trains policies over a family of simulators parameterized by latent dynamics parameters, typically sampled once per episode, to improve sim-to-real transfer [Tobin et al., 2017, Peng et al., 2018]. Uniform DR (UDR) remains a strong baseline and has enabled impressive transfer in complex robotic manipulation and locomotion [Tan et al., 2018, Akkaya et al., 2019]. Conceptually, DR can be viewed as learning in a latent-context MDP where the context (dynamics parameters) is constant within an episode but unobserved by the agent [Hallak et al., 2015, Modi et al., 2018]. This perspective highlights a core limitation of UDR: it implicitly commits to a hand-specified prior over deployment conditions, so both average performance and failure-mode coverage can be sensitive to the chosen randomization ranges.

Learning and adapting the randomization distribution. A natural extension is to adapt the sampling distribution over dynamics parameters rather than fixing it *a priori*. One line of work uses real-world data to infer plausible simulator-parameter distributions (system identification / posterior inference), thereby narrowing randomization to realistic regions [Ramos et al., 2019, Muratore et al., 2021]. Another line adapts the distribution using training signals in a curriculum/self-paced manner, expanding difficulty as the policy improves [Mehta et al., 2020, Klink et al., 2020, 2021]. Recent sampler-learning methods make the optimization explicit by combining a return-driven objective with coverage regularizers (e.g., entropy), often with additional constraints or trust regions to stabilize updates [Tiboni et al., 2024, Curtis et al., 2025]. Our work connects these adaptive-sampling ideas through a single *tilted-family* lens: the same exponential reweighting mechanism can express both right-tail (curriculum) and left-tail (robust) sampling, with the sign of a single temperature parameter determining which regime is emphasized.

Robust and risk-sensitive RL. Robust RL traditionally targets worst-case performance via minimax formulations over models or adversarial disturbances, yielding strong guarantees but often conservative solutions in continuous control [Wiesemann et al., 2013, Pinto et al., 2017b, Tessler et al., 2019, Kamalaruban et al., 2020]. Risk-sensitive RL instead optimizes tail- or utility-based criteria (e.g., CVaR or exponential-utility/entropic objectives), providing a principled robustness–performance trade-off [Mihatsch and Neuneier, 2002, Tamar et al., 2015, Prashanth and Michael, 2022, Foffano et al., 2025]. Soft-robust RL formalizes this idea by averaging over models while penalizing distribution shift, connecting risk sensitivity and robustness through an explicit divergence budget [Derman et al., 2018, Zhang et al., 2024]. In contrast to approaches that place risk on trajectory returns (or per-step disturbances), we place risk directly on the episode-level distribution over latent *dynamics parameters*, which is the uncertainty induced by domain randomization.

Distributional robustness and robust MDPs. Distributionally robust optimization (DRO) models uncertainty through ambiguity sets of probability distributions, often constructed around a reference distribution using ϕ-divergences or the Wasserstein metric [Namkoong and Duchi, 2016, Mohajerin Esfahani and Kuhn, 2018]. In sequential decision-making, classical robust and distributionally robust MDPs often exploit rectangular ambiguity over transition kernels to obtain dynamic programming solutions [Iyengar, 2005, Xu and Mannor, 2010, Wiesemann et al., 2013, Kumar et al., 2023, 2025]. Related stochastic control formulations address ambiguity in disturbance distributions using dynamic programming, with out-of-sample performance guarantees under Wasserstein ambiguity [Yang, 2021]. Uncertainty in noise distributions has also been studied in partially observed linear-quadratic control [Taskesen et al., 2023, Hakobyan and Yang, 2024]. In contrast, our ambiguity set is defined over the *episode-level* distribution of latent dynamics parameters, with each sampled parameter held fixed throughout an episode. This induces structured, generally non-rectangular coupling across time. Our practical method uses KL regularization over this distribution and learns the resulting adversarial reweighting as a sampler via variational inference, enabling policy-gradient training from simulator rollouts.

3 Problem setup and preliminaries

Parameterized MDPs for domain randomization. We consider continuous-control tasks modeled as a family of Markov decision processes (MDPs) parameterized by a vector of dynamics parameters $\xi \in \Xi \subseteq \mathbb{R}^d$. Each environment instance is an MDP $\mathcal{M}_\xi = (\mathcal{S}, \mathcal{A}, p_\xi, r, \mu, \gamma)$, where $\mathcal{S} \subseteq \mathbb{R}^{d_s}$ and $\mathcal{A} \subseteq \mathbb{R}^{d_a}$ denote the state and action spaces, $\gamma \in (0,1)$ is the discount factor, and $p_\xi(\cdot|s,a)$ is the transition kernel induced by ξ. The reward function $r(s,a)$ is assumed to be bounded. The initial-state distribution μ is assumed to be independent of ξ for simplicity; the formulation extends to ξ-dependent μ_ξ under the same measurability assumptions used to ensure $\xi \mapsto J(\theta;\xi)$ is well-defined (Appendix A.1).

The dynamics parameters ξ represent simulator quantities such as link masses, friction coefficients, actuator gains, and joint damping. In **domain randomization** (DR) [Tobin et al., 2017, Peng et al., 2018], an episode begins by sampling $\xi \sim \nu$ from a distribution over Ξ, and the sampled parameter is held fixed for the entire episode. The policy does *not* observe ξ directly; equivalently, the underlying process can be viewed as an MDP on the augmented state (s, ξ) with episode-constant ξ, which induces a latent-context (partially observed) MDP from the agent's perspective.[2]

Note that this modeling choice differs from robust-MDP formulations that place uncertainty directly on one-step transition kernels at each state-action pair; here, uncertainty enters through the episode-level distribution of latent dynamics parameters.

Domain-randomized objectives. Let π_θ be a stochastic policy parameterized by θ (possibly history-dependent); with slight abuse of notation we write $\pi_\theta(a \mid s)$ for brevity. For a fixed dynamics realization ξ, we define the discounted expected return

$$J(\theta;\xi) := \mathbb{E}_{\tau\sim\rho_{\theta,\xi}}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right], \tag{1}$$

[2]Accordingly, our formulation allows history-dependent (e.g., recurrent) policies when needed.

where $\tau := (s_0, a_0, s_1, a_1, \dots)$, and $\rho_{\theta,\xi}(\tau)$ denotes the state-action trajectory distribution induced by $s_0 \sim \mu$, $a_t \sim \pi_\theta(\cdot \mid s_t)$, and $s_{t+1} \sim p_\xi(\cdot \mid s_t, a_t)$. Since rewards are bounded and $\gamma \in (0,1)$, $J(\theta;\xi)$ is uniformly bounded for all ξ; see Appendix A.1.

Given a sampling distribution ν over dynamics parameters, the standard DR objective is to maximize the return averaged over ξ:

$$J_{\mathrm{DR}}(\theta, \nu) := \mathbb{E}_{\xi\sim\nu}\big[J(\theta;\xi)\big]. \tag{2}$$

Uniform domain randomization (UDR) corresponds to choosing ν as a fixed uniform distribution $\mathrm{Unif}(\Xi)$ over a user-specified bounded domain Ξ. In this case, robustness is assessed only relative to a fixed user-specified sampling distribution, which may under-emphasize adverse but important regions of the latent dynamics space.

Learning a sampler over dynamics parameters. Rather than fixing the distribution over dynamics parameters *a priori*, we learn a sampler over ξ jointly with the policy. We introduce a parametric family of distributions $\{\nu_\phi\}_{\phi\in\Phi}$ on Ξ and *alternately* update the policy and sampler. At iteration t, we sample a batch of dynamics parameters $\{\xi_i\}_{i=1}^{B} \sim \nu_{\phi_t}$, collect rollouts under π_θ in environments $\mathcal{M}_{\xi_i}$, update the policy using on-policy RL (we use PPO [Schulman et al., 2017]), and then update the sampler using the same rollouts.

We also define a reference distribution ν_0 over Ξ. In our experiments, we use $\nu_0 = \mathrm{Unif}(\Xi)$ for fair comparison with standard uniform domain randomization (UDR), though other priors (e.g., from system identification) could be incorporated.

4 Distributionally robust domain randomization

In this section, we present **risk-sensitive domain randomization (RSDR)** from a *distributionally robust* perspective. Rather than assuming a fixed training distribution over dynamics parameters ξ, we consider robustness against the worst-case distribution within a KL neighborhood of a reference distribution ν_0. This leads to a *soft adversary* over dynamics parameters, which we approximate using a learned sampler via variational inference. The practical algorithm uses the fixed-temperature KL-regularized objective described in Section 4.2.

4.1 Distributionally robust objective over dynamics parameters

Standard domain randomization optimizes the policy under a fixed sampling distribution ν over dynamics parameters (Eq. (2)). In sim-to-real settings, however, the true deployment distribution over ξ is typically unknown. To account for this uncertainty, we adopt a distributionally robust objective that evaluates each policy π_θ against the worst-case distribution within a KL neighborhood of a reference distribution ν_0:

$$J_{\mathrm{DRO}}(\theta) := \min_{\nu\in\mathcal{U}(\nu_0,c)} \mathbb{E}_{\xi\sim\nu}\big[J(\theta;\xi)\big], \tag{3}$$

where the ambiguity set is chosen as $\mathcal{U}(\nu_0, c) := \{\nu \ll \nu_0 : D_{\mathrm{KL}}(\nu \,\|\, \nu_0) \le c\}$.

Here $c > 0$ controls the size of the ambiguity set and ν_0 anchors the perturbations (e.g., $\nu_0 = \mathrm{Unif}(\Xi)$ in our experiments). Since $J(\theta;\xi)$ is uniformly bounded (Appendix A.1), the objective in Eq. (3) is well-defined for any $\nu \ll \nu_0$.

The corresponding fixed-radius policy-optimization problem is to learn a policy that maximizes the robust objective, i.e., $\max_\theta J_{\mathrm{DRO}}(\theta)$. Equivalently, this is the following distributionally robust optimization problem:

$$\max_\theta \min_{\nu\in\mathcal{U}(\nu_0,c)} \mathbb{E}_{\xi\sim\nu}\big[J(\theta;\xi)\big]. \tag{4}$$

Importantly, Eq. (4) models distributional robustness with respect to *episode-level* latent dynamics parameters ξ sampled once per episode. This induces a coupled (generally non-rectangular) perturbation of the transition dynamics across time and state–action pairs, in contrast to distributionally robust MDP formulations that impose (often rectangular) ambiguity sets over $p(\cdot \mid s, a)$ [Iyengar, 2005, Xu and Mannor, 2010, Wiesemann et al., 2013]; see Appendix A.6 for discussion. In our setting, uncertainty in ξ induces a coupled shift in the trajectory dynamics across time, making robustness over the latent parameter distribution the natural formulation for domain randomization.

4.2 Soft adversary via exponential tilting

A key property of KL-distributional robustness is that the minimization over ν in Eq. (3) admits a closed-form "soft adversary" solution. This result characterizes an adversarial reweighting of the probability mass over dynamics parameters and identifies the target distribution that our sampler aims to approximate.

Proposition 4.1 (Soft adversary under KL ambiguity)**.** *Fix θ and define $J(\xi) := J(\theta; \xi)$. For $\beta \leq 0$, define the exponentially tilted distribution*

$$\nu_\beta(d\xi) = \frac{1}{Z_\beta} \exp\big(\beta\, J(\xi)\big)\nu_0(d\xi), \tag{5}$$

where $Z_\beta = \mathbb{E}_{\xi\sim\nu_0}[\exp(\beta J(\xi))]$. Let $c_{\max} := \sup_{\beta<0} D_{\mathrm{KL}}(\nu_\beta\|\nu_0) \in [0,\infty]$. If $0 < c < c_{\max}$, then the inner minimization in (4) *admits an optimizer of the form $\nu^\star = \nu_{\beta_c}$ for some finite $\beta_c < 0$. Moreover, if J is not ν_0-a.s. constant, then $\beta_c < 0$ is unique and satisfies $D_{\mathrm{KL}}(\nu_{\beta_c}\|\nu_0) = c$ (i.e., the KL constraint is tight).*

Its proof is provided in Appendix A.2–A.4.

Remark 4.2. If $c = 0$ then the optimizer is ν_0 (corresponding to $\beta = 0$). If $c \geq c_{\max}$, the inner optimal value equals $J_- := \operatorname{ess\,inf}_{\xi\sim\nu_0} J(\xi)$ (hard worst-case) and is achieved in the limit $\beta \to -\infty$; see Appendix A.4–A.5 for details.

Scope of Proposition 4.1. The exponential-tilting form is a standard consequence of KL ambiguity and Donsker–Varadhan duality; our theoretical role for this result is to instantiate it for episode-wise domain randomization rather than to claim a new DRO duality. In our setting, the random variable being reweighted is the latent dynamics parameter ξ sampled once per episode. This induces a structured, generally non-rectangular perturbation of the trajectory distribution, which differs from robust-MDP formulations that use state-action rectangular transition uncertainty. This specialization identifies the policy-dependent target distribution that the learned dynamics sampler must track.

Proposition 4.1 shows that KL-distributional robustness induces a *soft* adversary over dynamics parameters: for $\beta < 0$, the tilted distribution ν_β places increasing probability mass on low-return (failure-prone) dynamics. As $\beta \to 0$, ν_β recovers the reference distribution ν_0 (Appendix A.5), while more negative values of β correspond to stronger robustness. Whenever $0 < c < c_{\max}$ and J is not ν_0-a.s. constant, the KL constraint is active and the corresponding $\beta_c < 0$ is unique; we provide tightness/uniqueness conditions and the $c \leftrightarrow \beta$ relationship in Appendix A.3–A.4.

Practical parametrization. For a fixed policy θ such that $J(\theta;\cdot)$ is not ν_0-a.s. constant, the KL radius and the temperature are in one-to-one correspondence on the interior range $0 < c < c_{\max}$: for every such c, there exists a unique finite $\beta_c < 0$ satisfying $D_{\mathrm{KL}}(\nu_{\beta_c}\|\nu_0) = c$ (Appendix A.4). In principle, one could enforce a fixed radius c by repeatedly solving for β_c as θ changes. We instead fix $\beta < 0$ and view it as the Lagrange temperature of the KL-regularized inner problem, corresponding to maximizing the entropic-risk objective

$$\mathcal{R}_\beta(\theta) := \frac{1}{\beta}\log \mathbb{E}_{\xi\sim\nu_0}\left[\exp\big(\beta J(\theta;\xi)\big)\right] = \min_{\nu \ll \nu_0}\Big\{\mathbb{E}_{\xi\sim\nu}[J(\theta;\xi)] - \tfrac{1}{\beta} D_{\mathrm{KL}}(\nu\|\nu_0)\Big\},$$

whose minimizer is exactly $\nu_\beta(\cdot;\theta)$. Consequently, the induced radius $c_\theta(\beta) = D_{\mathrm{KL}}(\nu_\beta(\cdot;\theta)\|\nu_0)$ can vary with the policy. Our implementation should therefore be read as a temperature-controlled entropic-risk/KL-DRO surrogate, rather than an exact fixed-radius solver. This avoids a noisy nested root-finding loop and gives a single continuous knob for the robustness strength. Appendix B gives the symmetric $\beta > 0$ optimistic counterpart, which we use only as an interpretive connection to curriculum DR.

4.3 Variational sampler learning

Proposition 4.1 characterizes the exact soft adversary for a *fixed* policy π_θ: the target is the exponentially tilted distribution $\nu_\beta(\cdot;\theta) \propto \exp(\beta J(\theta;\xi))\nu_0(\xi)$ over latent dynamics parameters. In practice, however, this target distribution changes with the policy during training, and direct sampling from ν_β is challenging because the normalizing constant $Z_\beta = \mathbb{E}_{\nu_0}[\exp(\beta J(\theta;\xi))]$ depends on the current policy and the return $J(\theta;\xi)$ is only available through black-box rollouts. To obtain a practical

training procedure, we introduce a learnable sampler $\nu_\phi(\xi)$ and fit it to the policy-induced tilted target distribution.

As the default instantiation of RSDR, we optimize ν_ϕ by minimizing the reverse KL divergence to the tilted target:

$$\mathcal{L}_{\mathrm{RKL}}(\phi;\theta,\beta) := D_{\mathrm{KL}}(\nu_\phi \,\|\, \nu_\beta)\,. \tag{6}$$

The reverse-KL objective in Eq. (6) contains the additive term $\log Z_\beta$, which is independent of ϕ. Dropping this term gives the equivalent optimization problem

$$\min_\phi \; \mathbb{E}_{\xi\sim\nu_\phi}\big[\log \nu_\phi(\xi) - \log \nu_0(\xi) - \beta J(\theta;\xi)\big]. \tag{7}$$

For $\nu_0 = \mathrm{Unif}(\Xi)$, the $\log \nu_0$ term is constant on Ξ, and the optimization problem in Eq. (7) is equivalent to

$$\max_\phi \; H(\nu_\phi) + \beta \mathbb{E}_{\xi\sim\nu_\phi}[J(\theta;\xi)], \tag{8}$$

where H denotes the differential entropy of the dynamics sampler, not the policy entropy. For $\beta < 0$, the second term favors low-return dynamics while the entropy term prevents premature collapse. For $\beta = 0$, the objective reduces to entropy maximization: the unrestricted optimum is the uniform reference $\nu_0 = \mathrm{Unif}(\Xi)$. For $\beta > 0$, the same objective becomes an optimistic entropy-regularized sampler related to self-paced curriculum objectives (Appendix C). RSDR uses the same variational fitting mechanism across $\beta > 0$ (curriculum), $\beta = 0$ (UDR target), and $\beta < 0$ (robust), differing only in the scalar β and the chosen sampler family; see Figure 2.

A direct solution of the full KL-DRO objective would require repeatedly solving the inner adversarial distribution problem for each policy iterate. MCMC [Brooks et al., 2011] could in principle sample from the policy-dependent tilted distribution $\nu_\beta(\xi;\theta) \propto \exp(\beta J(\theta;\xi))\nu_0(\xi)$ without knowing its normalizing constant, but it would still require many rollout-based evaluations of $J(\theta;\xi)$ and would need to re-equilibrate as the policy (and hence the target distribution) changes during training.

RSDR instead uses an alternating approximation: the parametric sampler ν_ϕ is updated online to track $\nu_\beta(\cdot;\theta)$ via the reverse-KL objective, while θ is updated using standard on-policy RL under the current sampler. Since the same rollout batch provides both policy-gradient estimates and return estimates for sampler learning, RSDR avoids a separate inner-loop sampling procedure at each policy iterate.

This tracking view is illustrated empirically in a toy example in Section 5.1, where the target distribution evolves with the policy and the learned sampler must adapt accordingly. In our implementation, the policy update is performed with PPO, and the complete procedure is summarized in Algorithm 1. We focus on the reverse-KL instantiation in the main text, using GMMVI [Arenz et al., 2023] as a practical reverse-KL-based sampling algorithm. Alternative sampler objectives and parameterizations are discussed in Appendix D. Full-training sampler-tracking diagnostics on two MuJoCo Playground tasks are provided in Appendix E.2.1.

5 Experiments

5.1 Mechanistic illustration with a toy example

Before turning to full continuous-control benchmarks, we first isolate the sampler-learning mechanism in a toy MDP where the policy-induced tilted target is analytically available. The purpose of this example is not to evaluate control performance, but to show that RSDR requires tracking a target distribution that changes as the policy changes. This distinction is important: standard variational inference [Blei et al., 2017] fits a fixed target, whereas RSDR repeatedly adapts the sampler toward the current tilted distribution induced by the policy.

Let p_u denote the probability of selecting an unsafe action under the current policy. Given this policy, we define a policy-induced tilted target over domain parameters, $\nu_\beta(\xi \mid \pi) \propto \exp\big(\beta d\, p_u\, g(\xi;\pi)\big)\big)$, where d is the dimension of the domain parameter and the dimension-dependent scaling is specific to the synthetic targets (Appendix E.1). Thus, changes in the policy induce corresponding changes in the target domain distribution. The learned sampler is therefore not solving a one-time fitting problem; rather, it must continually track a moving target over training.

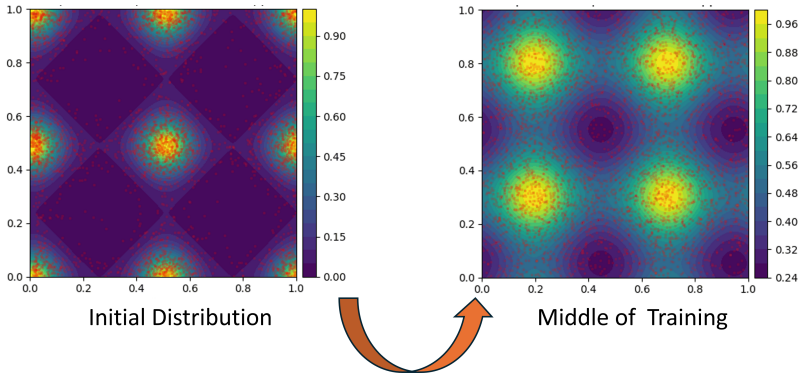


Figure 3: Toy illustration of a policy-induced target distribution with a shifting cosine mean. As the policy evolves during training, the target distribution moves accordingly.

Figure 3 illustrates this mechanism in a representative case where the target distribution shifts as the policy improves. The visualization shows that RSDR does not fit a single static adversarial distribution once and for all; instead, it updates the sampler to follow the current policy-induced tilted target.

The toy MDP therefore supports the mechanistic part of the RSDR story: the sampler must track a policy-dependent tilted distribution rather than fit a static domain distribution. Detailed definitions of the toy MDP, target families, and simulation results are provided in Appendix E.1.

5.2 Experimental setup

We evaluate RSDR on six MuJoCo Playground tasks, AcrobotSwingup, CartpoleSwingup, CheetahRun, WalkerWalk, PandaOpenCabinet, and PandaPickCubeOrientation [Zakka et al., 2025] under episode-wise dynamics randomization. A latent dynamics parameter $\xi \in \Xi$ is sampled once per episode and held fixed throughout the rollout.

To evaluate robustness across randomized dynamics, we report the mean return, minimum return, and CVaR_{10} over the return distribution induced by the dynamics parameters. Here, CVaR_{10} denotes the average of the worst 10% returns over evaluated domain parameters. Unless otherwise stated, results are reported as mean $\pm$ standard deviation over 10 independent training seeds.

We compare RSDR against risk-neutral uniform domain randomization (UDR) [Tobin et al., 2017], curriculum-style domain randomization baselines including AutoDR/ADR-style methods [Akkaya et al., 2019, Mehta et al., 2020], DORAEMON [Tiboni et al., 2024], and GoFlow [Curtis et al., 2025], as well as the robust RL method EPOpt [Rajeswaran et al., 2016]. Sampler details for RSDR are provided in Appendix D. Full task configurations, hyperparameter choices, and implementation details are provided in Appendices F and G. Code is available at `https://github.com/jeongsukchul/risk-sensitive-domain-randomization`.

5.3 Main benchmark results

We now evaluate whether the proposed risk-sensitive sampler improves robustness on standard domain-randomized control benchmarks. In Figure 4, across the evaluated tasks, risk-averse settings ($\beta < 0$) improve left-tail robustness, as measured by minimum return and CVaR_{10}. In contrast, risk-seeking settings ($\beta > 0$) tend to degrade left-tail robustness by concentrating sampling mass on high-return regions, as further illustrated in Appendix E.2 (see Table 4 and Table 5).

Overall, the benchmark results show that RSDR improves tail robustness more consistently than fixed domain randomization and curriculum-style baselines, while maintaining competitive mean performance in most settings. This tail improvement involves a task-dependent trade-off with mean return: on WalkerWalk, moving from $\beta = -10$ to $\beta = -30$ raises the minimum return from 362.36 to 432.03 but lowers the mean from 853.59 to 794.82, with the best CVaR_{10} at the intermediate $\beta = -20$. Across all six tasks, $\beta = -20$ improves CVaR_{10} over UDR by 6.7%–24.0%

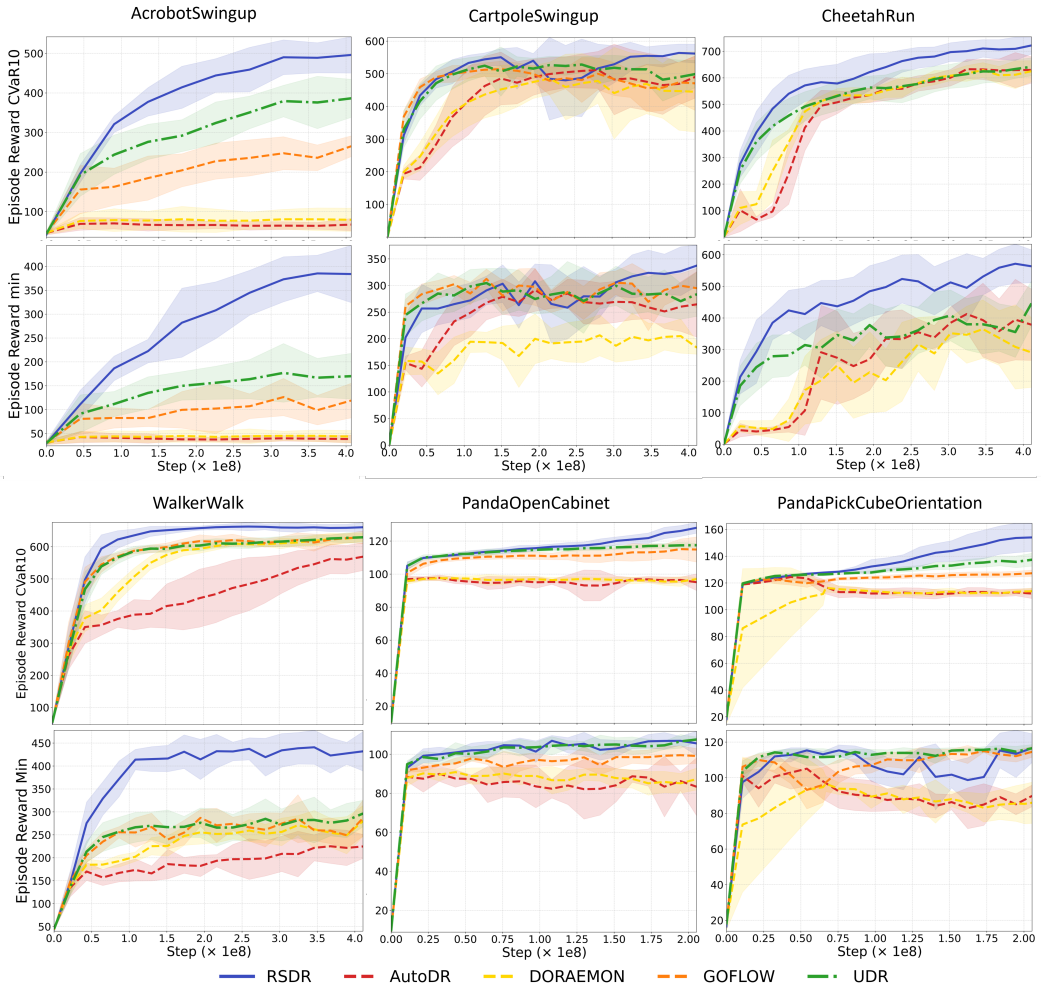


Figure 4: Robust evaluation over dynamics parameters Ξ with CVaR_{10} and minimum values. Risk-averse RSDR ($\beta < 0$) improves tail robustness compared to UDR and curriculum-oriented sampler learning baselines. Shaded regions denote ± 1 std over 10 training seeds.

with mean-return changes of -2.3% to $+5.1\%$, so we use it as a practical default, while $\beta = -10$ tends to better preserve the mean and $\beta = -30$ emphasizes the minimum more strongly (Section E.2).

We further evaluate a proof-of-concept realized-radius controller that adaptively adjusts β to approximately track a prescribed KL budget. Extended tables, full β-sweeps, adaptive-β results, and additional visualizations with sampler diagnostics are provided in Appendix E.

5.4 Comparison with EPOpt

We further compare RSDR against EPOpt [Rajeswaran et al., 2016], a classical robust RL baseline that updates the policy using only the worst-performing ϵ-fraction of sampled rollouts. Since EPOpt is sensitive to the choice of ϵ, we sweep multiple quantile thresholds and report the best-performing configuration for each task and metric.

Table 1 summarizes the comparison on CVaR_{10} and minimum return across the four tasks evaluated with EPOpt. RSDR achieves stronger tail performance than both UDR and the best-tuned EPOpt configuration on these metrics, suggesting that learning a soft risk-sensitive sampler over domain parameters is more effective than restricting policy updates to the worst-performing trajectory subset alone.

Table 1: Comparison with EPOpt on CVaR_{10} and minimum return.

Method	AcrobotSwingup	CartpoleSwingup	CheetahRun	WalkerWalk
		CVaR_{10}		
UDR	386.42 $\pm$ 48.05	499.04 $\pm$ 41.93	641.75 $\pm$ 33.99	629.43 $\pm$ 15.26
Best EPOpt (ϵ: .9/.8/.4/.7)	360.84 $\pm$ 88.85	548.44 $\pm$ 26.35	697.49 $\pm$ 62.73	637.63 $\pm$ 11.96
RSDR (Ours) (β: -30/-20/-30/-20)	**496.01** $\pm$ 45.81	**570.84** $\pm$ 19.43	**722.14** $\pm$ 32.45	**671.70** $\pm$ 10.07
		Minimum Return		
UDR	170.02 $\pm$ 47.82	283.61 $\pm$ 41.58	446.00 $\pm$ 71.00	296.47 $\pm$ 28.78
Best EPOpt (ϵ: .9/.8/.4/.8)	178.76 $\pm$ 57.95	304.59 $\pm$ 15.96	484.97 $\pm$ 77.23	308.38 $\pm$ 33.25
RSDR (Ours) (β: -30)	**383.93** $\pm$ 59.77	**337.47** $\pm$ 38.53	**563.43** $\pm$ 52.16	**432.03** $\pm$ 42.81

Note that EPOpt requires approximately B/ϵ rollouts per iteration to obtain B trajectories from the worst ϵ fraction for policy optimization, which increases the computational burden. In contrast, RSDR uses all sampled trajectories for both policy learning and sampler adaptation while still biasing training toward adverse domain configurations.

Consistent with this difference, wall-clock measurements show that RSDR incurs less than 2% additional training time relative to UDR on both profiled tasks, compared with 5.50%–40.38% for EPOpt under matched effective PPO-update batch sizes. Full EPOpt sweeps and computational overhead comparisons are provided in Appendix E.3.

5.5 Evaluation under non-stationary attacks

As an additional robustness stress test, we evaluate whether the robustness learned by RSDR under stationary episode-wise domain randomization transfers to environments whose dynamics drift within an episode. We consider two non-stationary perturbation processes: a projected Gaussian random walk driven by zero-mean Gaussian increments before projection, and a projected Langevin-type process that steers the domain parameters toward vulnerable regions of the learned sampler.

As shown in Figure 10, RSDR generally matches or outperforms UDR under both perturbation types, especially in CVaR_{10}. These results suggest that robustness learned from stationary episode-wise DR can transfer to temporally evolving dynamics. Full protocol details and additional visualizations are provided in Appendix E.4.

6 Concluding remarks

We proposed Risk-Sensitive Domain Randomization (RSDR), a distributionally robust formulation of episodic domain randomization over dynamics-parameter distributions. RSDR turns the KL-regularized soft adversary into a policy-dependent tilted target and learns an amortized dynamics sampler from rollout returns. Across continuous-control benchmarks, robust RSDR improves left-tail metrics over UDR and adaptive DR baselines, and outperforms best-tuned EPOpt.

This work has three main limitations. First, although the temperature parameter β provides a single tilted-family view of robust, neutral, and optimistic domain sampling, our experiments focus on the risk-averse regime ($\beta < 0$). Although we characterize the optimistic counterpart and evaluate realized-radius feedback, schedules that transition from optimistic to robust sampling remain unexplored; such schedules may improve exploration or curriculum learning and remain future work. Second, the default algorithm fixes β and performs amortized alternating updates between the sampler and the policy. It should therefore be viewed as a temperature-controlled KL-DRO surrogate rather than an exact solver of a fixed-radius minimax problem at every policy iterate. Third, our evaluation is simulation-based. Robustness to simulator-parameter variation is aligned with the mechanism of domain randomization, but it does not replace physical validation under simulator bias, sensing errors, actuator limits, and other unmodeled effects; moreover, choosing a realistic randomization range Ξ remains a modeling assumption shared with DR methods. More systematic adaptive temperature selection, larger randomization spaces, and evaluation on physical robotic systems remain important directions for future work.

Acknowledgments and Disclosure of Funding

This work was supported in part by the Information and Communications Technology Planning and Evaluation (IITP) grants funded by MSIT No. 2022-0-00124, No. 2022-0-00480 and No. RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University), the National Research Foundation of Korea (NRF) grants funded by MSIT No. RS-2026-25477173, No. RS-2026-25615715 and No. RS-2026-25504174, and the National IT Industry Promotion Agency (NIPA) grant funded by MSIT No. RS-2026-25621677.

References

Abbas Abdolmaleki, Rudolf Lioutikov, Jan R Peters, Nuno Lau, Luis Pualo Reis, and Gerhard Neumann. Model-based relative entropy stochastic search. *Advances in Neural Information Processing Systems*, 28, 2015.

Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

Oleg Arenz, Philipp Dahlinger, Zihan Ye, Michael Volpp, and Gerhard Neumann. A unified perspective on natural gradient variational inference with gaussian mixture models. *Transaction on Machine Learning Research*, 2023.

Julius Berner, Lorenz Richter, and Karen Ullrich. An optimal control perspective on diffusion-based generative modeling. *arXiv preprint arXiv:2211.01364*, 2022.

DJ Best and Nicholas I Fisher. Efficient simulation of the von mises distribution. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(2):152–157, 1979.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Denis Blessing, Xiaogang Jia, Johannes Esslinger, Francisco Vargas, and Gerhard Neumann. Beyond elbos: A large-scale evaluation of variational methods for sampling. In *International Conference on Machine Learning*, 2024.

Denis Blessing, Julius Berner, Lorenz Richter, Carles Domingo-Enrich, Yuanqi Du, Arash Vahdat, and Gerhard Neumann. Trust region constrained measure transport in path space for stochastic optimal control and inference. *arXiv preprint arXiv:2508.12511*, 2025.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of markov chain monte carlo*. CRC press, 2011.

Aidan Curtis, Eric Li, Michael Noseworthy, Nishad Gothoskar, Sachin Chitta, Hui Li, Leslie Pack Kaelbling, and Nicole E Carey. Flow-based domain randomization for learning and sequencing robotic skills. In *Forty-second International Conference on Machine Learning*, 2025.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Esther Derman, Daniel J Mankowitz, Timothy A Mann, and Shie Mannor. Soft-robust actor-critic policy-gradient. In *Association for Uncertainty in Artificial Intelligence*, 2018.

Haoyu Dong, Zhengmao He, Yang Li, Zhibin Li, Xinyu Yi, and Zhe Zhao. Closing the reality gap: Zero-shot sim-to-real deployment for dexterous force-based grasping and manipulation. *arXiv preprint arXiv:2601.02778*, 2026.

Monroe D Donsker and SR Srinivasa Varadhan. Asymptotic evaluation of certain markov process expectations for large time, i. *Communications on Pre and Applied Mathematics*, 28(1):1–47, 1975.

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. *Advances in Neural Information Processing Systems*, 32, 2019.

Jonas Eschmann, Dario Albani, and Giuseppe Loianno. Learning to fly in seconds. *IEEE Robotics and Automation Letters*, 9(7):6336–6343, 2024.

Daniele Foffano, Alessio Russo, and Alexandre Proutiere. Adversarial diffusion for robust reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Gerald B Folland. *Real analysis: modern techniques and their applications*. John Wiley & Sons, 1999.

Hans Föllmer and Thomas Knispel. Entropic risk measures: Coherence vs. convexity, model ambiguity and robust large deviations. *Stochastics and Dynamics*, 11(02n03):333–351, 2011.

C Daniel Freeman, Erik Frey, Anton Raichuk, Sertan Girgin, Igor Mordatch, and Olivier Bachem. Brax–a differentiable physics engine for large scale rigid body simulation. *arXiv preprint arXiv:2106.13281*, 2021.

Ignat Georgiev, Krishnan Srinivasan, Jie Xu, Eric Heiden, and Animesh Garg. Adaptive horizon actor-critic for policy learning in contact-rich differentiable simulation. In *International Conference on Machine Learning*, pages 15418–15437, 2024.

Mathis Gerdes and Miranda C. N. Cheng. Bijx: Bijections and normalizing flows with jax/nnx. *Journal of Open Source Software*, 10(116):9521, 2025. doi: 10.21105/joss.09521. URL `https://doi.org/10.21105/joss.09521`.

Vineet Goyal and Julien Grand-Clement. Robust markov decision process: Beyond rectangularity. *arXiv preprint arXiv:1811.00215*, 2018.

Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.

Astghik Hakobyan and Insoon Yang. Wasserstein distributionally robust control of partially observable linear stochastic systems. *IEEE Transactions on Automatic Control*, 69(9):6121–6136, 2024.

Assaf Hallak, Dotan Di Castro, and Shie Mannor. Contextual markov decision processes. *arXiv preprint arXiv:1502.02259*, 2015.

Chii-Ruey Hwang. Laplace's method revisited: weak convergence of probability measures. *The Annals of Probability*, pages 1177–1182, 1980.

Garud N Iyengar. Robust dynamic programming. *Mathematics of Operations Research*, 30(2): 257–280, 2005.

Parameswaran Kamalaruban, Yu-Ting Huang, Ya-Ping Hsieh, Paul Rolland, Cheng Shi, and Volkan Cevher. Robust reinforcement learning via adversarial training with langevin dynamics. *Advances in Neural Information Processing Systems*, 33:8127–8138, 2020.

Pascal Klink, Carlo D'Eramo, Jan R Peters, and Joni Pajarinen. Self-paced deep reinforcement learning. *Advances in Neural Information Processing Systems*, 33:9216–9227, 2020.

Pascal Klink, Hany Abdulsamad, Boris Belousov, Carlo D'Eramo, Jan Peters, and Joni Pajarinen. A probabilistic interpretation of self-paced learning with applications to reinforcement learning. *Journal of Machine Learning Research*, 22(182):1–52, 2021.

Navdeep Kumar, Esther Derman, Matthieu Geist, Kfir Y Levy, and Shie Mannor. Policy gradient for rectangular robust Markov decision processes. *Advances in Neural Information Processing Systems*, 36:59477–59501, 2023.

Navdeep Kumar, Adarsh Gupta, Maxence Mohamed Elfatihi, Giorgia Ramponi, Kfir Yehuda Levy, and Shie Mannor. Non-rectangular robust mdps with normed uncertainty sets. In *Advances in Neural Information Processing Systems*, 2025.

Quentin Le Lidec, Wilson Jallet, Louis Montaut, Ivan Laptev, Cordelia Schmid, and Justin Carpentier. Contact models in robotics: a comparative analysis. *IEEE Transactions on Robotics*, 2024.

Toru Lin, Kartik Sachdev, Linxi Fan, Jitendra Malik, and Yuke Zhu. Sim-to-real reinforcement learning for vision-based dexterous manipulation on humanoids. *arXiv preprint arXiv:2502.20396*, 2025.

Guan-Horng Liu, Jaemoo Choi, Yongxin Chen, Benjamin Kurt Miller, and Ricky TQ Chen. Adjoint schr\" odinger bridge sampler. *arXiv preprint arXiv:2506.22565*, 2025.

Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

Bhairav Mehta, Manfred Diaz, Florian Golemo, Christopher J Pal, and Liam Paull. Active domain randomization. In *Conference on Robot Learning*, pages 1162–1176. PMLR, 2020.

Oliver Mihatsch and Ralph Neuneier. Risk-sensitive reinforcement learning. *Machine Learning*, 49 (2):267–290, 2002.

Mayank Mittal, Pascal Roth, James Tigue, Antoine Richard, Octi Zhang, Peter Du, Antonio Serrano-Muñoz, Xinjie Yao, René Zurbrügg, Nikita Rudin, et al. Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning. *arXiv preprint arXiv:2511.04831*, 2025.

Aditya Modi, Nan Jiang, Satinder Singh, and Ambuj Tewari. Markov decision processes with continuous side information. In *Algorithmic Learning Theory*, pages 597–618. PMLR, 2018.

Peyman Mohajerin Esfahani and Daniel Kuhn. Data-driven distributionally robust optimization using the wasserstein metric: Performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1):115–166, 2018.

Fabio Muratore, Christian Eilers, Michael Gienger, and Jan Peters. Data-efficient domain randomization with bayesian optimization. *IEEE Robotics and Automation Letters*, 6(2):911–918, 2021.

Hongseok Namkoong and John C Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. *Advances in Neural Information Processing Systems*, 29, 2016.

Rhys Newbury, Jack Collins, Kerry He, Jiahe Pan, Ingmar Posner, David Howard, and Akansel Cosgun. A review of differentiable simulators. *IEEE Access*, 2024.

Nikolas Nüsken and Lorenz Richter. Solving high-dimensional hamilton–jacobi–bellman pdes using neural networks: perspectives from the theory of controlled diffusions and measures on path space. *Partial differential equations and applications*, 2(4):48, 2021.

Sindhu Padakandla. A survey of reinforcement learning algorithms for dynamically varying environments. *ACM Computing Surveys*, 54(6):1–25, 2021.

Victor M Panaretos and Yoav Zemel. Statistical aspects of wasserstein distances. *Annual review of statistics and its application*, 6(1):405–431, 2019.

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in Neural Information Processing Systems*, 30, 2017.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021.

Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *IEEE International Conference on Robotics and Automation*, pages 3803–3810. IEEE, 2018.

Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. Asymmetric actor critic for image-based robot learning. *arXiv preprint arXiv:1710.06542*, 2017a.

Lerrel Pinto, James Davidson, Rahul Sukthankar, and Abhinav Gupta. Robust adversarial reinforcement learning. In *International Conference on Machine Learning*, pages 2817–2826. PMLR, 2017b.

LA Prashanth and C Fu Michael. Risk-sensitive reinforcement learning via policy gradient search. *Foundations and Trends in Machine Learning*, 15(5):537–693, 2022.

Aravind Rajeswaran, Sarvjeet Ghotra, Balaraman Ravindran, and Sergey Levine. Epopt: Learning robust neural network policies using model ensembles. *arXiv preprint arXiv:1610.01283*, 2016.

Fabio Ramos, Rafael Carvalhaes Possas, and Dieter Fox. Bayessim: adaptive domain randomization via probabilistic inference for robotics simulators. *arXiv preprint arXiv:1906.01728*, 2019.

Lorenz Richter and Julius Berner. Improved sampling via learned diffusions. *arXiv preprint arXiv:2307.01198*, 2023.

Lorenz Richter, Ayman Boustati, Nikolas Nüsken, Francisco Ruiz, and Omer Deniz Akyildiz. Vargrad: a low-variance gradient estimator for variational inference. *Advances in Neural Information Processing Systems*, 33:13481–13492, 2020.

Walter Rudin. *Real and complex analysis*. McGraw-Hill, Inc., 1987.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Hyung Ju Suh, Max Simchowitz, Kaiqing Zhang, and Russ Tedrake. Do differentiable simulators give better policy gradients? In *International Conference on Machine Learning*, pages 20668–20696, 2022.

Aviv Tamar, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. Policy gradient for coherent risk measures. *Advances in Neural Information Processing Systems*, 28, 2015.

Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. *arXiv preprint arXiv:1804.10332*, 2018.

Bahar Taskesen, Dan Iancu, Çağıl Koçyiğit, and Daniel Kuhn. Distributionally robust linear quadratic control. *Advances in Neural Information Processing Systems*, 36:18613–18632, 2023.

Chen Tessler, Yonathan Efroni, and Shie Mannor. Action robust reinforcement learning and applications in continuous control. In *International Conference on Machine Learning*, pages 6215–6224. PMLR, 2019.

Gabriele Tiboni, Pascal Klink, Jan Peters, Tatiana Tommasi, Carlo D'Eramo, and Georgia Chalvatzaki. Domain randomization via entropy maximization. In *International Conference on Learning Representations*, 2024.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 23–30. IEEE, 2017.

S. R. S. Varadhan. Asymptotic probabilities and differential equations. *Communications on Pure and Applied Mathematics*, 19:261–286, 1966.

Wolfram Wiesemann, Daniel Kuhn, and Berç Rustem. Robust markov decision processes. *Mathematics of Operations Research*, 38(1):153–183, 2013.

Eliot Xing, Vernon Luk, and Jean Oh. Stabilizing reinforcement learning in differentiable multiphysics simulation. In *International Conference on Learning Representations*, 2025.

Huan Xu and Shie Mannor. Distributionally robust markov decision processes. *Advances in Neural Information Processing Systems*, 23, 2010.

Jie Xu, Viktor Makoviychuk, Yashraj Narang, Fabio Ramos, Wojciech Matusik, Animesh Garg, and Miles Macklin. Accelerated policy learning with parallel differentiable simulation. In *International Conference on Learning Representations*, 2021.

Insoon Yang. Wasserstein distributionally robust stochastic control: A data-driven approach. *IEEE Transactions on Automatic Control*, 66(8):3863–3870, 2021.

Kevin Zakka, Baruch Tabanpour, Qiayuan Liao, Mustafa Haiderbhai, Samuel Holt, Jing Yuan Luo, Arthur Allshire, Erik Frey, Koushil Sreenath, Lueder A Kahrs, et al. Mujoco playground. *arXiv preprint arXiv:2502.08844*, 2025.

Runyu Zhang, Yang Hu, and Na Li. Soft robust MDPs and risk-sensitive MDPs: Equivalence, policy gradient, and sample complexity. In *International Conference on Learning Representations*, 2024.

Adil Zouitine, David Bertoin, Pierre Clavier, Matthieu Geist, and Emmanuel Rachelson. Time-constrained robust mdps. *Advances in Neural Information Processing Systems*, 37:35574–35611, 2024.

A KL-ambiguity DRO over dynamics-parameter distributions: Full derivations

This appendix provides the derivations underlying Eq. (5) in Proposition 4.1, and clarifies boundary cases. Throughout, we fix a policy parameter θ and write $J(\xi) := J(\theta; \xi)$ for brevity. We treat ν_0 and ν as probability measures on $(\Xi, \mathcal{B}(\Xi))$; when densities exist (e.g., $\nu_0 = \mathrm{Unif}(\Xi)$), the measure statements reduce to density formulas.

A.1 Regularity of $J(\theta; \xi)$ in our setting

Recall that the reward is bounded, i.e., $|r(s, a)| \le r_{\max}$ for all (s, a), and $\gamma \in (0, 1)$. Therefore, for any θ and any ξ, $|J(\theta; \xi)| \le \frac{r_{\max}}{1-\gamma}$. Consequently, assuming measurability of $\xi \mapsto J(\theta; \xi)$,

$$J(\theta; \cdot) \in L^\infty(\Xi, \mathcal{B}(\Xi), \nu_0),$$

i.e., the return is essentially bounded with respect to any reference measure ν_0.

Remark A.1 (Measurability). Measurability of $\xi \mapsto J(\theta; \xi)$ holds under standard assumptions on the simulator family, e.g., when $(s, a, \xi) \mapsto p_\xi(\cdot \mid s, a)$ defines a jointly measurable Markov kernel and r, π_θ are measurable. In many physics simulators used for domain randomization, the dependence on continuous parameters (e.g., masses, frictions, gains) can be modeled as Borel measurable [Rudin, 1987] in ξ; contact may introduce discontinuities, but measurability typically still holds under mild conditions (e.g., piecewise continuous dependence on ξ), which we assume throughout.

A.2 Primal inner minimization problem under a KL-ambiguity set

Recall the KL-ambiguity set around a reference distribution ν_0: $\mathcal{U}(\nu_0, c) := \{\nu \ll \nu_0 : D_{\mathrm{KL}}(\nu \| \nu_0) \le c\}$, $c > 0$. Because $\nu \ll \nu_0$, we can represent ν via the Radon–Nikodym derivative

$$w(\xi) := \frac{d\nu}{d\nu_0}(\xi),$$

which satisfies $w(\xi) \ge 0$ ν_0-a.s. and $\mathbb{E}_{\nu_0}[w] = 1$. Moreover, for any measurable f, $\mathbb{E}_\nu[f(\xi)] = \mathbb{E}_{\nu_0}[w(\xi) f(\xi)]$. Using the convention $0 \log 0 := 0$, we have

$$D_{\mathrm{KL}}(\nu \| \nu_0) = \mathbb{E}_\nu \left[\log \frac{d\nu}{d\nu_0} \right] = \mathbb{E}_{\nu_0}[w \log w].$$

Thus, Eq. (3) is equivalent to the following convex program (for fixed θ):

$$\min_{w \ge 0} \ \mathbb{E}_{\nu_0}[w(\xi) J(\xi)] \quad \text{s.t.} \quad \mathbb{E}_{\nu_0}[w] = 1, \ \ \mathbb{E}_{\nu_0}[w \log w] \le c. \tag{9}$$

Problem (9) is convex since the objective is linear in w and $w \mapsto w \log w$ is convex. Slater's condition [Boyd and Vandenberghe, 2004] holds for $c > 0$ since $w \equiv 1$ satisfies $\mathbb{E}_{\nu_0}[w] = 1$ and strictly satisfies $\mathbb{E}_{\nu_0}[w \log w] = 0 < c$; therefore, strong duality applies.

Lemma A.2 (Existence of an optimal worst-case distribution)**.** *Suppose that $J \in L^\infty(\nu_0)$ and $c \in (0, \infty)$. Then, the primal problem* (9) *attains its minimum; equivalently, the inner minimization in* (4) *admits an optimizer $\nu^\star \in \mathcal{U}(\nu_0, c)$.*

Proof sketch. Let

$$\mathcal{F} := \big\{ w \in L^1(\nu_0) : \ w \ge 0, \ \mathbb{E}_{\nu_0}[w] = 1, \ \mathbb{E}_{\nu_0}[w \log w] \le c \big\}.$$

Take a minimizing sequence $\{w_n\} \subset \mathcal{F}$. Define $\Phi(x) := x \log x - x + 1$, which is convex, nonnegative, and satisfies $\Phi(x)/x \to \infty$. Since $\mathbb{E}[w_n] = 1$ and $\mathbb{E}[w_n \log w_n] \le c$, we have

$$\sup_n \mathbb{E}_{\nu_0}[\Phi(w_n)] < \infty.$$

By the de la Vallée–Poussin criterion [Folland, 1999], $\{w_n\}$ is uniformly integrable. Thus, it follows from the Dunford–Pettis theorem [Folland, 1999] that $\{w_n\}$ is relatively weakly compact in $L^1(\nu_0)$. Therefore, (along a subsequence) $w_n \rightharpoonup w^\star$ weakly in $L^1(\nu_0)$.

The constraints $\mathbb{E}[w] = 1$ and $w \geq 0$ pass to the limit. Moreover, the functional $w \mapsto \mathbb{E}_{\nu_0}[w \log w]$ is weakly lower semicontinuous on $L^1(\nu_0)$. Thus, $\mathbb{E}_{\nu_0}[w^\star \log w^\star] \leq \liminf_n \mathbb{E}_{\nu_0}[w_n \log w_n] \leq c$, yielding $w^\star \in \mathcal{F}$.

Since $J \in L^\infty(\nu_0)$, the objective $w \mapsto \mathbb{E}_{\nu_0}[wJ]$ is a continuous linear functional on $L^1(\nu_0)$, hence weakly continuous. Therefore,

$$\mathbb{E}_{\nu_0}[w^\star J] = \lim_{n\to\infty} \mathbb{E}_{\nu_0}[w_n J] = \inf_{w\in\mathcal{F}} \mathbb{E}_{\nu_0}[wJ],$$

so the minimum is attained at $w^\star$.

Finally, define a probability measure $\nu^\star$ on $(\Xi, \mathcal{B}(\Xi))$ by

$$\nu^\star(A) := \int_A w^\star(\xi)\, \nu_0(d\xi), \qquad A \in \mathcal{B}(\Xi).$$

Then $\nu^\star \ll \nu_0$, $D_{\mathrm{KL}}(\nu^\star \| \nu_0) = \mathbb{E}_{\nu_0}[w^\star \log w^\star] \leq c$. Therefore, $\nu^\star \in \mathcal{U}(\nu_0, c)$, and it attains the inner minimum in (4). □

A.3 KKT derivation of the exponentially tilted worst-case distribution ($\beta < 0$)

Consider the reformulated problem (9) for evaluating $J_{\mathrm{DRO}}(\theta)$. Introduce Lagrange multipliers $\lambda \geq 0$ for the KL constraint and $\eta \in \mathbb{R}$ for the normalization constraint. We first consider the regime in which the optimal KL multiplier satisfies $\lambda > 0$. By complementary slackness, this implies that the KL constraint is tight. The case $\lambda = 0$ is discussed in Appendix A.5.

The Lagrangian of (9) is

$$\mathcal{L}(w, \lambda, \eta) = \mathbb{E}_{\nu_0}[wJ] + \lambda\big(\mathbb{E}_{\nu_0}[w \log w] - c\big) + \eta\big(\mathbb{E}_{\nu_0}[w] - 1\big). \tag{10}$$

Since the Lagrangian is an expectation of a separable convex integrand in $w(\xi)$, the minimization over w decomposes pointwise over ξ. For $\lambda > 0$, the pointwise minimizer satisfies that for ν_0-a.e. ξ,

$$J(\xi) + \lambda(\log w(\xi) + 1) + \eta = 0. \tag{11}$$

which yields

$$w(\xi) = C\, \exp\Big(-\frac{1}{\lambda} J(\xi)\Big), \tag{12}$$

where $C = \exp(-1 - \eta/\lambda)$ is a constant independent of ξ. Enforcing $\mathbb{E}_{\nu_0}[w] = 1$ determines C:

$$C^{-1} = \mathbb{E}_{\nu_0}\left[\exp\Big(-\frac{1}{\lambda} J(\xi)\Big)\right]. \tag{13}$$

Define the temperature parameter

$$\beta := -\frac{1}{\lambda}. \tag{14}$$

Since $\lambda > 0$, we have $\beta < 0$. The resulting worst-case distribution ν_β satisfies

$$\frac{d\nu_\beta}{d\nu_0}(\xi) = \frac{\exp\big(\beta J(\xi)\big)}{Z_\beta}, \qquad Z_\beta = \mathbb{E}_{\xi\sim\nu_0}\left[\exp\big(\beta J(\xi)\big)\right], \tag{15}$$

i.e.,

$$\nu_\beta(d\xi) = \frac{1}{Z_\beta} \exp\big(\beta J(\xi)\big)\, \nu_0(d\xi), \qquad \beta < 0. \tag{16}$$

This establishes the exponential-tilting form in Proposition 4.1.

Density form (when applicable). If ν_0 admits a density (still denoted by $\nu_0(\xi)$) with respect to Lebesgue measure [Rudin, 1987] on Ξ (e.g., $\nu_0 = \mathrm{Unif}(\Xi)$ on a bounded domain), then ν_β also admits a density and (16) becomes

$$\nu_\beta(\xi) = \frac{1}{Z_\beta} \exp\big(\beta J(\xi)\big)\, \nu_0(\xi), \qquad Z_\beta = \int_\Xi \exp\big(\beta J(\xi)\big)\, \nu_0(\xi)\, d\xi.$$

Constraint activity and tightness. Let $\nu^\star$ be a primal optimizer and let $\lambda^\star \geq 0$ be an associated optimal multiplier for the KL constraint. By complementary slackness,

$$\lambda^\star\big(D_{\mathrm{KL}}(\nu^\star\|\nu_0) - c\big) = 0.$$

Thus, $\lambda^\star > 0$ implies that the KL constraint is tight, while a slack KL constraint implies $\lambda^\star = 0$.

To make the attainable KL range explicit, define

$$c_{\max} := \sup_{\beta<0} D_{\mathrm{KL}}(\nu_\beta\|\nu_0) \in [0,\infty].$$

When J is not ν_0-a.s. constant and $0 < c < c_{\max}$, there exists a unique finite $\beta_c < 0$ such that $D_{\mathrm{KL}}(\nu_{\beta_c}\|\nu_0) = c$. In this interior regime, $\nu^\star = \nu_{\beta_c}$ and $\lambda^\star = -1/\beta_c > 0$. The uniqueness of β_c follows from the strict monotonicity of $c(\beta) := D_{\mathrm{KL}}(\nu_\beta\|\nu_0)$ on $\beta < 0$ when J is not ν_0-a.s. constant (Appendix A.4).

At the boundary $c = c_{\max} < \infty$, the KL constraint can be tight even when $\lambda^\star = 0$, so tightness alone does not guarantee a finite negative temperature. If $c \geq c_{\max}$, the inner optimal value coincides with the hard worst-case value; see Appendix A.5 and Remark A.3.

A.4 Monotone relationship between c and β

Recall that the tilted distribution is defined (Appendix A.3) by

$$\nu_\beta(d\xi) = \frac{\exp(\beta J(\xi))}{Z_\beta}\,\nu_0(d\xi), \qquad Z_\beta := \mathbb{E}_{\nu_0}\big[\exp(\beta J)\big].$$

Let $A(\beta) := \log Z_\beta = \log \mathbb{E}_{\nu_0}[\exp(\beta J)]$. Since $J \in L^\infty(\nu_0)$, we have $Z_\beta < \infty$ for all β. Then, differentiation under the expectation is justified by dominated convergence because $\exp(\beta J)$ and $J\exp(\beta J)$ are dominated by integrable constants for any fixed β. Hence A is twice differentiable and

$$A'(\beta) = \mathbb{E}_{\nu_\beta}[J], \qquad A''(\beta) = \mathrm{Var}_{\nu_\beta}(J) \geq 0. \tag{17}$$

Using $\log \frac{d\nu_\beta}{d\nu_0} = \beta J - A(\beta)$, we obtain

$$D_{\mathrm{KL}}(\nu_\beta\|\nu_0) = \mathbb{E}_{\nu_\beta}\left[\log\frac{d\nu_\beta}{d\nu_0}\right] = \mathbb{E}_{\nu_\beta}[\beta J - A(\beta)] = \beta\,\mathbb{E}_{\nu_\beta}[J] - A(\beta). \tag{18}$$

Recall that $c(\beta) := D_{\mathrm{KL}}(\nu_\beta\|\nu_0)$. Differentiating (18) and using (17) yields

$$c'(\beta) = \beta\,\mathrm{Var}_{\nu_\beta}(J). \tag{19}$$

If J is not ν_0-a.s. constant, then $\mathrm{Var}_{\nu_\beta}(J) > 0$ for every finite β. Indeed, for any finite β,

$$\frac{d\nu_\beta}{d\nu_0}(\xi) = \frac{\exp(\beta J(\xi))}{Z_\beta} > 0 \quad \nu_0\text{-a.e.},$$

so ν_β is equivalent to ν_0 and J cannot become ν_β-a.s. constant unless it is ν_0-a.s. constant. Therefore, $c(\beta)$ is strictly decreasing on $(-\infty, 0)$ and strictly increasing on $(0,\infty)$.

Finally, recall

$$c_{\max} := \lim_{\beta\to-\infty} c(\beta) = \sup_{\beta<0} c(\beta) \in [0,\infty].$$

Since $c(\beta) \to 0$ as $\beta \to 0^-$ (equivalently $\nu_\beta \to \nu_0$; see Appendix A.5), it follows that for any $c \in (0, c_{\max})$ there exists a unique $\beta_c < 0$ such that $D_{\mathrm{KL}}(\nu_{\beta_c}\|\nu_0) = c$. Equivalently, the inverse map $c \mapsto \beta_c$ is strictly decreasing on $(0, c_{\max})$.

Remark A.3 (When is $c_{\max}$ finite?). Let $S := \{\xi : J(\xi) = J_-\}$ where $J_- = \operatorname{ess\,inf}_{\xi\sim\nu_0} J(\xi)$. Suppose first $\nu_0(S) > 0$, and define the normalized restriction $\nu_0(A \mid S) := \nu_0(A \cap S)/\nu_0(S)$. Then $\nu_\beta \to \nu_0(\cdot \mid S)$ in total variation as $\beta \to -\infty$ (see Lemma A.4 below), and

$$c_{\max} = \lim_{\beta\to-\infty} D_{\mathrm{KL}}(\nu_\beta\|\nu_0) = D_{\mathrm{KL}}(\nu_0(\cdot \mid S)\|\nu_0) = \log\frac{1}{\nu_0(S)} < \infty.$$

In particular, for any $c \geq c_{\max}$, the conditional distribution $\nu_0(\cdot \mid S)$ is feasible and optimal for (3), so the KL constraint may be slack when $c > c_{\max}$. If $\nu_0(S) = 0$, then $c_{\max} = +\infty$; see Corollary A.6 in Appendix A.5.

Lemma A.4 (Limit of the tilted distribution when $\nu_0(S) > 0$)**.** *Let $J_- := \operatorname{ess\,inf}_{\nu_0} J$ and $S := \{\xi : J(\xi) = J_-\}$. If $\nu_0(S) > 0$, then $\nu_\beta \to \nu_0(\cdot \mid S)$ in total variation as $\beta \to -\infty$.*

Proof. Write $J(\xi) = J_- + \Delta(\xi)$ with $\Delta(\xi) \geq 0$ ν_0-a.s. and $\Delta(\xi) = 0$ on S. Then for any $\beta < 0$,

$$\frac{d\nu_\beta}{d\nu_0}(\xi) = \frac{e^{\beta J(\xi)}}{\mathbb{E}_{\nu_0}[e^{\beta J}]} = \frac{e^{\beta \Delta(\xi)}}{\mathbb{E}_{\nu_0}[e^{\beta \Delta}]} =: w_\beta(\xi).$$

Since $0 \leq e^{\beta\Delta(\xi)} \leq 1$ and $e^{\beta\Delta(\xi)} \to \mathbf{1}_S(\xi)$ pointwise as $\beta \to -\infty$, dominated convergence yields $\mathbb{E}_{\nu_0}[e^{\beta\Delta}] \to \mathbb{E}_{\nu_0}[\mathbf{1}_S] = \nu_0(S)$. For sufficiently negative β, $\mathbb{E}_{\nu_0}[e^{\beta\Delta}] \geq \nu_0(S)/2$, so $w_\beta(\xi) \leq 2/\nu_0(S)$. Moreover, $w_\beta(\xi) \to \mathbf{1}_S(\xi)/\nu_0(S)$ pointwise as $\beta \to -\infty$. Since

$$\left| w_\beta(\xi) - \frac{\mathbf{1}_S(\xi)}{\nu_0(S)} \right| \leq w_\beta(\xi) + \frac{1}{\nu_0(S)} \leq \frac{3}{\nu_0(S)},$$

another dominated convergence argument gives $w_\beta \to \mathbf{1}_S/\nu_0(S)$ in $L^1(\nu_0)$. Because both ν_β and $\nu_0(\cdot \mid S)$ are absolutely continuous w.r.t. ν_0 with densities w_β and $\mathbf{1}_S/\nu_0(S)$, respectively,

$$\|\nu_\beta - \nu_0(\cdot \mid S)\|_{\mathrm{TV}} = \frac{1}{2} \int \left| w_\beta - \frac{\mathbf{1}_S}{\nu_0(S)} \right| d\nu_0 = \frac{1}{2} \|w_\beta - \mathbf{1}_S/\nu_0(S)\|_1 \to 0.$$

□

A.5 Limiting behaviors

Define the essential extrema under the reference measure:

$$J_- := \operatorname*{ess\,inf}_{\xi \sim \nu_0} J(\xi), \qquad J_+ := \operatorname*{ess\,sup}_{\xi \sim \nu_0} J(\xi).$$

Note that $J \in L^\infty(\nu_0)$ implies $J_-, J_+ \in \mathbb{R}$.

$\beta \to 0$ recovers ν_0. Since $J \in L^\infty(\nu_0)$, dominated convergence gives $Z_\beta = \mathbb{E}_{\nu_0}[\exp(\beta J)] \to 1$ as $\beta \to 0$. Moreover, for any fixed $\beta_0 > 0$ and all $|\beta| \leq \beta_0$,

$$0 \leq \frac{d\nu_\beta}{d\nu_0}(\xi) = \frac{\exp(\beta J(\xi))}{Z_\beta} \leq \frac{\exp(\beta_0 \|J\|_\infty)}{\exp(-\beta_0 \|J\|_\infty)} = \exp(2\beta_0 \|J\|_\infty),$$

so $\frac{d\nu_\beta}{d\nu_0} \to 1$ in $L^1(\nu_0)$. Consequently, $\|\nu_\beta - \nu_0\|_{\mathrm{TV}} = \frac{1}{2} \int \left| \frac{d\nu_\beta}{d\nu_0} - 1 \right| d\nu_0 \to 0$, i.e., $\nu_\beta \to \nu_0$ in total variation.

$\beta \to -\infty$ approaches a hard worst-case limit (in value). Let $A_\varepsilon := \{\, \xi : J(\xi) \leq J_- + \varepsilon \,\}$. By definition of essential infimum, $\nu_0(A_\varepsilon) > 0$ for all $\varepsilon > 0$. For $\beta < 0$,

$$\nu_0(A_\varepsilon) \exp(\beta(J_- + \varepsilon)) \leq Z_\beta \leq \exp(\beta J_-).$$

Dividing by β and letting $\beta \to -\infty$ yields

$$\lim_{\beta \to -\infty} \frac{1}{\beta} \log Z_\beta = J_-$$

Concentration near J_-. Fix $\varepsilon > 0$ and define $A_{\varepsilon/2} := \{\xi : J(\xi) \leq J_- + \varepsilon/2\}$. By the definition of $J_- = \operatorname{ess\,inf}_{\nu_0} J$, we have $\nu_0(A_{\varepsilon/2}) > 0$. For $\beta < 0$,

$$\begin{aligned} \nu_\beta(J > J_- + \varepsilon) &= \frac{\int_{\{J > J_- + \varepsilon\}} \exp(\beta J(\xi))\, \nu_0(d\xi)}{Z_\beta} \leq \frac{\exp(\beta(J_- + \varepsilon))}{\int_{A_{\varepsilon/2}} \exp(\beta J(\xi))\, \nu_0(d\xi)} \\ &\leq \frac{\exp(\beta(J_- + \varepsilon))}{\nu_0(A_{\varepsilon/2})\, \exp(\beta(J_- + \varepsilon/2))} = \frac{1}{\nu_0(A_{\varepsilon/2})} \exp(\beta\varepsilon/2) \xrightarrow[\beta \to -\infty]{} 0. \end{aligned} \tag{20}$$

Since $J \in [J_-, J_+]$, we have $J_- \leq \mathbb{E}_{\nu_\beta}[J] \leq (J_- + \varepsilon) + (J_+ - J_-)\nu_\beta(J > J_- + \varepsilon)$, so letting $\beta \to -\infty$ and then $\varepsilon \downarrow 0$ yields $\mathbb{E}_{\nu_\beta}[J] \to J_-$.

Lemma A.5 (Information monotonicity of KL)**.** *For any probability measures P, Q on $(\mathcal{X}, \mathcal{F})$ and any measurable map $f : \mathcal{X} \to \mathcal{Y}$, we have*

$$D_{\mathrm{KL}}(P\|Q) \;\geq\; D_{\mathrm{KL}}(P \circ f^{-1} \,\|\, Q \circ f^{-1}).$$

Corollary A.6. *Let $S := \{\xi : J(\xi) = J_-\}$, where $J_- = \operatorname{ess\,inf}_{\xi\sim\nu_0} J(\xi)$. If $\nu_0(S) = 0$, then $\sup_{\beta<0} D_{\mathrm{KL}}(\nu_\beta\|\nu_0) = +\infty$, i.e., $c_{\max} = +\infty$.*

Proof. For $\varepsilon > 0$, let $A_\varepsilon := \{\xi : J(\xi) \leq J_- + \varepsilon\}$. Then $A_\varepsilon \downarrow \{J \leq J_-\}$ as $\varepsilon \downarrow 0$. Since $\nu_0(\{J < J_-\}) = 0$ by definition of J_- and $\nu_0(S) = 0$ by assumption, we have $\nu_0(\{J \leq J_-\}) = 0$. Therefore, by continuity from above, we have $\nu_0(A_\varepsilon) \to 0$.

Fix $\varepsilon > 0$. By the concentration bound (20), for all $\beta < 0$,

$$\nu_\beta(A_\varepsilon^c) \leq \frac{e^{\beta\varepsilon/2}}{\nu_0(A_{\varepsilon/2})}.$$

Choose $\beta = \beta(\varepsilon) < 0$ such that the RHS is at most $1/2$, so that $p := \nu_{\beta(\varepsilon)}(A_\varepsilon) \geq 1/2$. Let $q := \nu_0(A_\varepsilon)$.

Apply Lemma A.5 with $f(\xi) = \mathbf{1}_{A_\varepsilon}(\xi)$:

$$D_{\mathrm{KL}}(\nu_{\beta(\varepsilon)}\|\nu_0) \;\geq\; D_{\mathrm{KL}}(\mathrm{Bern}(p) \,\|\, \mathrm{Bern}(q)),$$

where $\mathrm{Bern}(p)$ denotes the Bernoulli distribution on $\{0, 1\}$ with $\mathbb{P}(1) = p$ and $\mathbb{P}(0) = 1 - p$. If $q \leq 1/2$, it follows from $p \geq 1/2$ that

$$D_{\mathrm{KL}}(\mathrm{Bern}(p) \,\|\, \mathrm{Bern}(q)) = p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q} \;\geq\; \frac{1}{2} \log \frac{1}{4q}.$$

Now take $\varepsilon_n \downarrow 0$. Since $q_n := \nu_0(A_{\varepsilon_n}) \to 0$, we obtain $D_{\mathrm{KL}}(\nu_{\beta(\varepsilon_n)}\|\nu_0) \to +\infty$. Therefore $\sup_{\beta<0} D_{\mathrm{KL}}(\nu_\beta\|\nu_0) = +\infty$. □

A.6 Remark: relation to classical distributionally robust MDP formulations

Classical robust and distributionally robust MDP formulations typically define ambiguity sets directly over the *one-step transition kernel* at each state–action pair, e.g., sets of $p(\cdot \mid s, a)$, often together with a (state–action-wise) *rectangularity* assumption that enables Bellman-style dynamic programming guarantees [Iyengar, 2005, Xu and Mannor, 2010, Wiesemann et al., 2013]. In contrast, our ambiguity set $\mathcal{U}(\nu_0, c)$ is defined over the *episode-level distribution of latent dynamics parameters* ξ, which are sampled once per episode and held fixed thereafter. This induces a coupled (generally non-rectangular) perturbation of the transition dynamics across time and across state–action pairs: conditioning on history changes the posterior over ξ, so the induced process is generally non-Markov in s alone (while it is Markov on the augmented state (s, ξ)). As a result, the standard rectangular robust-MDP machinery does not apply directly, and our formulation is naturally implemented via sampling-based (policy-gradient) training together with a learned dynamics-parameter sampler.

B Entropic-risk duality and symmetric optimistic counterpart

While the main text focuses on the risk-averse regime ($\beta < 0$) for robust domain randomization, the same exponentially tilted formulation also admits a symmetric optimistic counterpart for $\beta > 0$. This appendix presents that broader entropic-risk perspective Föllmer and Knispel [2011]. In particular, the sign of β determines whether the induced domain distribution emphasizes low-return or high-return regions, yielding a unified view of pessimistic and optimistic domain randomization within the same exponential family.

B.1 Dual representation as entropic risk

We begin with the KL-based distributionally robust objective and recall its dual representation in terms of an entropic-risk functional. The derivation combines strong duality for the primal problem in Appendix A.2 with the Donsker–Varadhan variational identity [Donsker and Varadhan, 1975, Varadhan, 1966].

Corollary B.1 (Dual representation as entropic risk)**.** *Fix θ and write $J(\xi) := J(\theta;\xi)$. Assume J is ν_0-measurable and essentially bounded (Appendix A.1), and let $c > 0$. Then the KL-based distributionally robust objective*

$$J_{\mathrm{DRO}}(\theta) := \min_{\nu:\, D_{\mathrm{KL}}(\nu\|\nu_0)\le c} \mathbb{E}_{\xi\sim\nu}[J(\xi)]$$

admits the dual representation

$$J_{\mathrm{DRO}}(\theta) = \sup_{\beta<0}\left\{\frac{1}{\beta}\log\mathbb{E}_{\xi\sim\nu_0}\left[\exp\big(\beta J(\xi)\big)\right] + \frac{c}{\beta}\right\}. \tag{21}$$

Moreover, whenever the dual optimum is attained at a finite temperature (equivalently, the associated KL multiplier satisfies $\lambda > 0$ so that $\beta = -1/\lambda \in (-\infty, 0)$), any primal optimizer must be of the exponentially tilted form

$$\nu_\beta(d\xi) \propto \exp(\beta J(\xi))\,\nu_0(d\xi).$$

If J is not ν_0-a.s. constant and c lies in the interior of the achievable KL range (e.g., $0 < c < \sup_{\beta<0} D_{\mathrm{KL}}(\nu_\beta\|\nu_0)$), then the KL constraint is tight, the maximizing $\beta_c < 0$ is unique, and so is the corresponding primal optimizer (Appendix A.4).

Proof. Consider the primal problem

$$\min_{\nu:\, D_{\mathrm{KL}}(\nu\|\nu_0)\le c} \mathbb{E}_{\xi\sim\nu}[J(\xi)].$$

By Appendix A.2, this is a convex optimization problem in the Radon–Nikodym derivative $w = \frac{d\nu}{d\nu_0}$ and satisfies Slater's condition(e.g., $w \equiv 1$ is strictly feasible for $c > 0$), so strong duality holds.

Introduce the Lagrange multiplier $\lambda \ge 0$ for the KL constraint. The Lagrange dual is

$$J_{\mathrm{DRO}}(\theta) = \sup_{\lambda\ge 0}\left\{\inf_{\nu\ll\nu_0}\Big(\mathbb{E}_\nu[J] + \lambda D_{\mathrm{KL}}(\nu\|\nu_0)\Big) - \lambda c\right\}. \tag{22}$$

For $\lambda > 0$, apply the Donsker–Varadhan variational identity

$$\log\mathbb{E}_{\nu_0}[\exp(f(\xi))] = \sup_{\nu\ll\nu_0}\left\{\mathbb{E}_{\xi\sim\nu}[f(\xi)] - D_{\mathrm{KL}}(\nu\|\nu_0)\right\}, \tag{23}$$

with $f(\xi) = -J(\xi)/\lambda$ to obtain

$$\sup_{\nu\ll\nu_0}\left\{\mathbb{E}_\nu\left[-\frac{1}{\lambda}J\right] - D_{\mathrm{KL}}(\nu\|\nu_0)\right\} = \log\mathbb{E}_{\nu_0}\left[\exp\left(-\frac{1}{\lambda}J\right)\right].$$

Since $\lambda > 0$, multiplying both sides by $-\lambda$ flips the supremum into an infimum, yielding

$$\inf_{\nu\ll\nu_0}\Big(\mathbb{E}_\nu[J] + \lambda D_{\mathrm{KL}}(\nu\|\nu_0)\Big) = -\lambda\log\mathbb{E}_{\nu_0}\left[\exp\left(-\frac{1}{\lambda}J\right)\right].$$

Substituting into (22) (restricted to $\lambda > 0$) gives

$$J_{\mathrm{DRO}}(\theta) = \sup_{\lambda>0}\left\{-\lambda\log\mathbb{E}_{\nu_0}\left[\exp\left(-\frac{1}{\lambda}J\right)\right] - \lambda c\right\}.$$

Now reparameterize with $\beta := -1/\lambda$, so that $\beta < 0$ and $\lambda = -1/\beta$. Then

$$-\lambda\log\mathbb{E}_{\nu_0}\left[\exp\left(-\frac{1}{\lambda}J\right)\right] = \frac{1}{\beta}\log\mathbb{E}_{\nu_0}[\exp(\beta J)], \qquad -\lambda c = \frac{c}{\beta}.$$

Hence,

$$J_{\mathrm{DRO}}(\theta) = \sup_{\beta<0}\left\{\frac{1}{\beta}\log\mathbb{E}_{\nu_0}[\exp(\beta J)] + \frac{c}{\beta}\right\}.$$

For the optimizer form: whenever the optimum is attained at $\lambda > 0$ (equivalently $\beta \in (-\infty, 0)$), the maximizer in (23) (with $f = -J/\lambda$) is the Gibbs/exponentially tilted measure, hence any primal optimizer must satisfy $\nu_\beta(d\xi) \propto \exp(\beta J(\xi))\,\nu_0(d\xi)$; see also the KKT derivation in Appendix A.3. □

Corollary B.2 (Dual attainment in the interior)**.** *Fix θ and assume $J(\cdot) = J(\theta;\cdot) \in L^\infty(\nu_0)$. Let $\nu_\beta(d\xi) \propto e^{\beta J(\xi)}\nu_0(d\xi)$ for $\beta < 0$, and define $c_{\max} := \sup_{\beta<0} D_{\mathrm{KL}}(\nu_\beta\|\nu_0) \in [0,\infty]$. If $0 < c < c_{\max}$, then there exists a unique $\beta_c < 0$ such that $D_{\mathrm{KL}}(\nu_{\beta_c}\|\nu_0) = c$ (Appendix A.4). Moreover, the dual supremum in* (21) *is attained at β_c, and*

$$J_{\mathrm{DRO}}(\theta) = \mathbb{E}_{\xi\sim\nu_{\beta_c}}[J(\xi)] = \frac{1}{\beta_c}\log\mathbb{E}_{\xi\sim\nu_0}\left[e^{\beta_c J(\xi)}\right] + \frac{c}{\beta_c}.$$

Proof. By the Donsker–Varadhan identity, for any $\beta < 0$ and any $\nu \ll \nu_0$,

$$\beta\mathbb{E}_\nu[J] - D_{\mathrm{KL}}(\nu\|\nu_0) \le \log\mathbb{E}_{\nu_0}[e^{\beta J}],$$

so dividing by $\beta < 0$ gives

$$\mathbb{E}_\nu[J] \ge \frac{1}{\beta}\log\mathbb{E}_{\nu_0}[e^{\beta J}] + \frac{1}{\beta}D_{\mathrm{KL}}(\nu\|\nu_0).$$

If $\nu \in \mathcal{U}(\nu_0, c)$, then $D_{\mathrm{KL}}(\nu\|\nu_0) \le c$ implies $\frac{1}{\beta}D_{\mathrm{KL}}(\nu\|\nu_0) \ge \frac{c}{\beta}$, hence

$$\mathbb{E}_\nu[J] \ge \frac{1}{\beta}\log\mathbb{E}_{\nu_0}[e^{\beta J}] + \frac{c}{\beta}.$$

Taking the infimum over $\nu \in \mathcal{U}(\nu_0, c)$ yields $J_{\mathrm{DRO}}(\theta) \ge \sup_{\beta<0}\{\frac{1}{\beta}\log\mathbb{E}_{\nu_0}[e^{\beta J}] + \frac{c}{\beta}\}$.

Now take $\beta_c < 0$ with $D_{\mathrm{KL}}(\nu_{\beta_c}\|\nu_0) = c$. Then ν_{β_c} is feasible. Moreover, equality holds in the Donsker–Varadhan bound at $\nu = \nu_{\beta_c}$, so

$$\mathbb{E}_{\nu_{\beta_c}}[J] = \frac{1}{\beta_c}\log\mathbb{E}_{\nu_0}[e^{\beta_c J}] + \frac{1}{\beta_c}D_{\mathrm{KL}}(\nu_{\beta_c}\|\nu_0) = \frac{1}{\beta_c}\log\mathbb{E}_{\nu_0}[e^{\beta_c J}] + \frac{c}{\beta_c}.$$

Therefore, $J_{\mathrm{DRO}}(\theta) \le \mathbb{E}_{\nu_{\beta_c}}[J]$ and the reverse inequality above forces equality. Thus the dual supremum is attained at β_c and the displayed identity holds. □

The previous two corollaries characterize the risk-averse regime used in the main text. We now record the symmetric optimistic counterpart obtained by reversing the role of low-return and high-return domain regions under the same KL ambiguity set.

B.2 Symmetric optimistic counterpart ($\beta > 0$)

Fix θ and write $J(\xi) := J(\theta;\xi)$. Consider the symmetric optimistic counterpart over the same ambiguity set:

$$J_{\mathrm{OPT}}(\theta) := \max_{\nu\in\mathcal{U}(\nu_0,c)} \mathbb{E}_{\xi\sim\nu}[J(\xi)]. \tag{24}$$

This is equivalent to minimizing $\mathbb{E}_\nu[-J]$ under the KL constraint, so the derivation in Appendix A.3 applies with J replaced by $-J$.

For $\beta \ge 0$, define the exponentially tilted family

$$\nu_\beta(d\xi) = \frac{1}{Z_\beta}\exp(\beta J(\xi))\,\nu_0(d\xi), \qquad Z_\beta := \mathbb{E}_{\xi\sim\nu_0}[\exp(\beta J(\xi))]. \tag{25}$$

Let $c^+_{\max} := \sup_{\beta>0} D_{\mathrm{KL}}(\nu_\beta\|\nu_0) \in [0,\infty]$.

Proposition B.3 (Soft optimist under KL ambiguity)**.** *Assume $J \in L^\infty(\nu_0)$ and $0 < c < c^+_{\max}$. Then the maximization problem* (24) *admits an optimizer of the form $\nu^\star = \nu_{\beta_c}$ for some finite $\beta_c \ge 0$. Moreover, if J is not ν_0-a.s. constant, then $\beta_c > 0$ is unique and satisfies $D_{\mathrm{KL}}(\nu_{\beta_c}\|\nu_0) = c$ (i.e., the KL constraint is tight).*

Remark B.4 (Boundary cases). If $c = 0$, then the unique optimizer is $\nu^\star = \nu_0$ (corresponding to $\beta = 0$). If $c \ge c^+_{\max}$, then the inner optimal value equals the hard best-case value $J_+ := \operatorname{ess\,sup}_{\xi\sim\nu_0} J(\xi)$ and is achieved in the limit $\beta \to +\infty$.

Proof sketch. Note that $\max_{\nu\in\mathcal{U}(\nu_0,c)} E_\nu[J] = -\min_{\nu\in\mathcal{U}(\nu_0,c)} E_\nu[-J]$. Since $J \in L^\infty(\nu_0)$, we have $-J \in L^\infty(\nu_0)$ as well, and the existence of an optimizer follows by applying Lemma A.2 to $-J$.

Next, apply the KKT derivation in Appendix A.3 with J replaced by $-J$. Whenever $\lambda > 0$, the optimizer must satisfy

$$w(\xi) \;\propto\; \exp\!\left(\frac{1}{\lambda}J(\xi)\right),$$

i.e., $\nu^\star = \nu_\beta$ for $\beta := 1/\lambda \geq 0$ as in (25).

Finally, define $c(\beta) := D_{\mathrm{KL}}(\nu_\beta\|\nu_0)$. Appendix A.5 shows $\nu_\beta \to \nu_0$ in total variation as $\beta \to 0^+$, hence $c(\beta) \to 0$. Appendix A.4 shows that $c(\beta)$ is continuous and strictly increasing on $(0,\infty)$ whenever J is not ν_0-a.s. constant. Thus, for any $c \in (0, c^+_{\max})$ there exists a unique $\beta_c > 0$ such that $c(\beta_c) = c$ (intermediate value theorem), and complementary slackness implies the KL constraint is tight at the optimum. □

Limit as $\beta \to +\infty$. Let $J_+ := \operatorname{ess\,sup}_{\xi\sim\nu_0} J(\xi)$. Since J is essentially bounded (Appendix A.1), $J_+ < \infty$ and $Z_\beta < \infty$ for all β. A standard log-sum-exp (Laplace) principle [Hwang, 1980] yields

$$\lim_{\beta\to+\infty} \frac{1}{\beta}\log \mathbb{E}_{\nu_0}[\exp(\beta J)] = J_+.$$

Moreover, for any $\varepsilon > 0$, letting $A_\varepsilon := \{J \geq J_+ - \varepsilon\}$ (so $\nu_0(A_\varepsilon) > 0$),

$$\nu_\beta(A_\varepsilon^c) = \frac{\mathbb{E}_{\nu_0}\left[e^{\beta J}\mathbf{1}_{A_\varepsilon^c}\right]}{\mathbb{E}_{\nu_0}\left[e^{\beta J}\right]} \leq \frac{e^{\beta(J_+-\varepsilon)}}{e^{\beta(J_+-\varepsilon/2)}\nu_0(A_{\varepsilon/2})} = \frac{1}{\nu_0(A_{\varepsilon/2})}e^{-\beta\varepsilon/2} \to 0.$$

Thus, $\nu_\beta(J < J_+ - \varepsilon) \to 0$ and consequently $\mathbb{E}_{\nu_\beta}[J] \to J_+$ as $\beta \to +\infty$.

Taken together, these results show that the risk-averse formulation used in the main text is one side of a broader exponential-tilting family indexed by β. Negative values of β emphasize low-return regions and induce a soft adversarial distribution, whereas positive values emphasize high-return regions and yield a symmetric optimistic counterpart. This broader perspective is not required for the main algorithmic development, but it clarifies how robust and optimistic domain randomization arise from the same underlying variational principle.

B.3 Relation to CVaR-based Adversarial Sampling

Recent CVaR-based adversarial sampling methods, such as AD-RRL [Foffano et al., 2025], reweight trajectories toward low-return outcomes. In contrast, RSDR applies KL-based reweighting to episode-level dynamics parameters.

For a common reference distribution ν_0, the CVaR risk envelope at level α(the lower-tail probability mass) is

$$\mathcal{U}^\alpha_{\mathrm{CVaR}} = \left\{\nu \ll \nu_0 : \frac{d\nu}{d\nu_0} \leq \frac{1}{\alpha}\right\}.$$

Since every distribution in this envelope satisfies $D_{\mathrm{KL}}(\nu\|\nu_0) \leq \log(1/\alpha)$, we have

$$\mathcal{U}^\alpha_{\mathrm{CVaR}} \subseteq \mathcal{U}^{\log(1/\alpha)}_{\mathrm{KL}} := \left\{\nu \ll \nu_0 : D_{KL}(\nu\|\nu_0) \leq \log(\frac{1}{\alpha})\right\}.$$

This suggests an envelope-based calibration: given α, set $c = \log(1/\alpha)$ and choose a policy-dependent temperature $\beta < 0$ satisfying

$$D_{\mathrm{KL}}(\nu_{\beta,\theta}\|\nu_0) = \log(1/\alpha),$$

whenever such a finite solution exists. For nonconstant $J(\theta;\cdot)$ and $0 < \log(1/\alpha) < c_{\max}$, this temperature is unique and can be obtained by one-dimensional root finding (Appendix A.4). The resulting KL-DRO objective is not generally equivalent to CVaR, but provides a related risk-sensitive formulation.

Randomization Method	Sampler family ν_ϕ	Update principle	Regime	Extra stabilizers
UDR	fixed uniform	none (fixed $\nu_\phi = \nu_0$)	neutral ($\beta = 0$)	none
AutoDR	box-uniform	success-gated	optimistic (implicit)	threshold-based gating
DORAEMON	(product) Beta	max-entropy with success-probability constraint	optimistic (conceptual)	KL trust region to ν_{old}
GoFlow	normalizing flow	self-paced / max-entropy objective	optimistic ($\beta > 0$)	regularization / trust region
RSDR (ours)	general VI family	fit $\nu_\beta \propto e^{\beta J}\nu_0$ via reverse-KL VI	all ($\beta \in \mathbb{R}$)	handled by VI optimizer

Table 2: Conceptual comparison of DR sampler-learning methods: UDR [Tobin et al., 2017, Peng et al., 2018], AutoDR [Akkaya et al., 2019], DORAEMON [Tiboni et al., 2024], GoFlow [Curtis et al., 2025], and RSDR. The regime column provides an interpretation rather than asserting algorithmic equivalence or a shared temperature parameter. Methods differ in their sampler families, performance statistics, and constraints or trust regions. In particular, DORAEMON constrains success probability, whereas RSDR fits a target tilted by expected return, $\nu_\beta \propto \exp(\beta J)\nu_0$. RSDR uses the same variational fitting principle across regimes, changing only β.

C Connection to self-paced and curriculum RL in the risk-seeking regime

The previous section established that $\beta > 0$ yields a symmetric optimistic counterpart to the robust $\beta < 0$ regime. In this section, we connect that optimistic exponential tilting to self-paced and curriculum-style learning. The key point is that when $\beta > 0$, the tilted distribution $\nu_\beta(\xi) \propto \exp(\beta J(\xi))\nu_0(\xi)$ emphasizes high-return regions of the domain space, which is conceptually aligned with curriculum methods that bias training toward easier or more favorable domains.

For $\beta > 0$, the RSDR sampler objective rewards high expected return while penalizing divergence from the reference distribution ν_0, providing a conceptual connection to self-paced and curriculum learning. This comparison does not imply algorithmic equivalence: the methods use different performance statistics and constraints. In particular, DORAEMON constrains the probability of task success, whereas RSDR uses expected return in its exponential tilt.

Curriculum learning organizes training by emphasizing easier or more favorable tasks before expanding toward broader coverage. In reinforcement learning, Self-Paced Reinforcement Learning (SPRL) [Klink et al., 2020, 2021] formalizes this idea as a distribution optimization problem over task or context variables in contextual MDPs [Hallak et al., 2015]. Recent methods such as DORAEMON [Tiboni et al., 2024] and GoFlow [Curtis et al., 2025] adapt this intuition to domain randomization by learning a training distribution over simulator parameters. The general Self-Paced Curriculum Learning objective is formulated as:

$$\max_{\pi,\nu} \ \mathbb{E}_{p(\mathbf{c}|\nu)}[J_\pi(\mathbf{c})] - \alpha D_{\mathrm{KL}}\big(p(\mathbf{c} \mid \nu) \,\|\, \mu(\mathbf{c})\big), \tag{26}$$

where $\mathbf{c} \in \mathcal{C}$ denotes the context or task variable, μ is a target distribution over contexts, and $p(\mathbf{c} \mid \nu)$ is the train-time context distribution parameterized by ν. The coefficient α trades off performance against proximity to the target distribution.

To automatically schedule the difficulty, SPRL methods typically solve a constrained optimization problem for the distribution update:

$$\min_{\boldsymbol{\nu}} \ D_{\mathrm{KL}}(p(\mathbf{c} \mid \boldsymbol{\nu}) \| \mu(\mathbf{c})) \tag{27a}$$

$$\text{s.t.} \ \ \mathbb{E}_{p(\mathbf{c}|\boldsymbol{\nu})}[J_\pi(\mathbf{c})] \geq J_T \tag{27b}$$

$$D_{\mathrm{KL}}\left(p(\mathbf{c} \mid \boldsymbol{\nu}) \| p\left(\mathbf{c} \mid \boldsymbol{\nu}'\right)\right) \leq \epsilon. \tag{27c}$$

DORAEMON [Tiboni et al., 2024] maximizes the entropy of the dynamics distribution subject to a minimum success probability and a KL trust region. For a fixed policy π_θ, let $I_{\mathrm{succ}}(\tau) \in \{0, 1\}$ indicate whether trajectory τ is successful, and define

$$s_\theta(\xi) := \mathbb{E}_{\tau \sim \rho_{\theta,\xi}}\big[I_{\mathrm{succ}}(\tau)\big].$$

For a return-threshold success criterion, $I_{\text{succ}}(\tau) = \mathbf{1}\{\widetilde{G}(\tau) \geq J_T\}$, where $\widetilde{G}(\tau)$ denotes the normalized episodic return. The sampler update is

$$\begin{aligned} \max_{\phi \in \Phi} \quad & \mathcal{H}(\nu_\phi) \\ \text{s.t.} \quad & \mathbb{E}_{\xi \sim \nu_\phi}[s_\theta(\xi)] \geq \alpha_D, \\ & D_{\text{KL}}(\nu_\phi \| \nu_{\text{old}}) \leq \epsilon_D. \end{aligned} \tag{28}$$

Here J_T determines trajectory success, α_D is the required success probability, and ϵ_D is the KL trust-region radius. Thus, the performance constraint in Eq. (28) concerns success probability rather than expected return. The task-specific thresholds and baseline hyperparameters are reported in Appendix G.2.

GoFlow [Curtis et al., 2025] utilizes Normalizing Flows [Papamakarios et al., 2021] to represent ν_ϕ and directly optimizes a regularized objective similar to Eq. (26):

$$\max_{\phi, \pi} \left\{ \mathbb{E}_{\xi \sim \nu_\phi} \left[J_\pi(\xi) \right] + \alpha \mathcal{H}(\nu_\phi) - \beta D_{KL} \left(\nu_{\text{old}} \| \nu_\phi \right) \right\}. \tag{29}$$

AutoDR [Akkaya et al., 2019] follows a similar principle without explicit optimization, using heuristic boundary expansion to maximize the entropy (volume) of a box distribution.

$$\max_{\phi \in \Phi} \mathcal{H}(\nu_\phi), \quad \text{where } \nu_\phi(\xi) = \prod_{i=1}^{d} U(\phi_i^L, \phi_i^H), \tag{30}$$

but it uses heuristic evaluations to expand the domain only when performance exceeds the threshold J_T.

From the relation between our RSDR objective (8) and the equations above, the $\beta > 0$ regime of RSDR is not merely a sign-reversal of the robust objective, but rather an optimistic reweighting mechanism closely related to self-paced and curriculum-style domain randomization, which typically introduce additional constraints or trust regions to stabilize learning. The main distinction is that RSDR expresses this behavior through the same variational fitting principle used in the robust regime, changing only the sign and magnitude of β. Table 2 summarizes how this interpretation relates RSDR to UDR, AutoDR, DORAEMON, and GoFlow.

D Sampler objectives and parameterizations

In the main text, RSDR is instantiated with a reverse-KL variational sampler as the default training procedure; see Section 4.3. This appendix provides additional implementation details for the choice(Gaussian Mixture Model Variational Inference), and then describes alternative sampler objectives(Log Variance) and parameterizations(Diffusion based parameterization) that we also explored. Our goal here is not to present these variants as co-equal components of the core method, but to document practical design choices and auxiliary alternatives beyond the main-text instantiation.

D.1 Default reverse-KL sampler: GMMVI

As described in the main text, our default instantiation of RSDR learns the sampler ν_ϕ by minimizing the reverse-KL objective. For this purpose, we parameterize ν_ϕ using Gaussian Mixture Model Variational Inference (GMMVI) [Arenz et al., 2023].

We adopt GMMVI for three practical reasons. First, in standard model-free RL, the gradient $\nabla_\xi J(\theta; \xi)$ is typically unavailable, since computing it would require differentiating through the transition dynamics $\mathcal{P}_\xi$. This issue is especially acute in rigid-body robotics, where contact dynamics are often stiff or non-differentiable [Suh et al., 2022, Le Lidec et al., 2024, Newbury et al., 2024]. Even when differentiable simulators are available, propagating gradients through long-horizon trajectories is often unstable in practice [Xu et al., 2021, Georgiev et al., 2024, Xing et al., 2025].

Second, we use a GMMVI variant based on Model-Based Relative Entropy Stochastic Search (MORE) [Abdolmaleki et al., 2015], which optimizes the reverse-KL objective using local quadratic regression models of the energy values rather than simulator gradients. Also, it uses information-geometric trust regions for careful local exploration that stabilize the training procedure. This makes

the sampler update compatible with black-box rollout evaluations while helping mitigate premature mode collapse.

Third, prior empirical comparisons suggest that GMMVI can be more robust to mode collapse and more sample-efficient than alternatives such as normalizing flows [Blessing et al., 2024]. In our implementation, we follow the practical design choices in the unified GMMVI framework of Arenz et al. [2023].

To respect the bounded parameter domain Ξ (assumed to be a hyper-rectangle), we map unconstrained sampler outputs through a $\tanh$ bijector and account for the transformation using the standard change-of-variables formula [Haarnoja et al., 2018, Papamakarios et al., 2021].

Our implementation builds upon the code provided at `https://github.com/DenisBless/variational_sampling_methods`. We introduced specific modifications to ensure full compatibility with JAX's Just-In-Time (JIT) compilation, thereby significantly accelerating the training process.

D.2 Alternative log-variance sampler: DIS-LV

Beyond the default reverse-KL instantiation used in the main text, we also explored a diffusion-based sampler trained with a log-variance objective, which we denote DIS-LV [Richter and Berner, 2023]. We treat this as an alternative sampler variant rather than a component of the core RSDR method. We include it both as a sampler-choice ablation in the toy-example setting and, more broadly, to emphasize that RSDR is not tied to a single variational family.

The main motivation for considering the log-variance objective is similar to that for GMMVI: it avoids requiring the intractable gradient $\nabla_\xi J(\theta;\xi)$. This makes it appealing in settings where returns are available only through black-box rollouts. In addition, log-variance objectives have been used in diffusion-based variational sampling as a practical alternative to direct reverse-KL optimization [Richter et al., 2020, Nüsken and Richter, 2021, Berner et al., 2022, Richter and Berner, 2023, Blessing et al., 2025].

As in the GMMVI case, we constrain the compact parameter space Ξ using a $\tanh$ bijector and account for the induced density transformation through the standard change-of-variables formula. Our implementation follows the public `sde_sampler` codebase `https://github.com/juliusberner/sde_sampler/tree/main/sde_sampler`, but reimplemented in JAX for efficiency.

In our toy-example ablations(Appendix E.1), DIS-LV was a competitive alternative, although it was generally less effective than the default reverse-KL GMMVI sampler on problems with sharp local modes. Accordingly, we retain reverse-KL GMMVI as the default instantiation in the main text and present DIS-LV here as an auxiliary alternative. The results for Benchmark RL tests are in Appendix E.5.

E Experimental details

E.1 Toy MDP for policy-induced exponentially tilted dynamics

We consider a minimal episodic MDP designed to expose the policy–environment coupling underlying Risk-Sensitive Domain Randomization (RSDR). The state space consists of an initial state S_0 and two terminal states, S_{good} and S_{bad}. Every episode starts at S_0, the agent selects one action, and the episode terminates immediately after transitioning to one of the two terminal states. The terminal rewards are

$$r(S_{\text{good}}) = 1, \qquad r(S_{\text{bad}}) = 0.$$

Hence the return of an episode is completely determined by whether the transition from S_0 lands in the good or bad terminal state.

The environment is indexed by a latent domain parameter $\xi \in [0,1]^d$, which plays the role of a randomized dynamics parameter. At the initial state S_0, the agent chooses between two actions: a safe action a_s and an unsafe action a_u. The safe action induces the transition

$$P(S_{\text{good}} \mid S_0, a_s) = q, \qquad P(S_{\text{bad}} \mid S_0, a_s) = 1 - q,$$

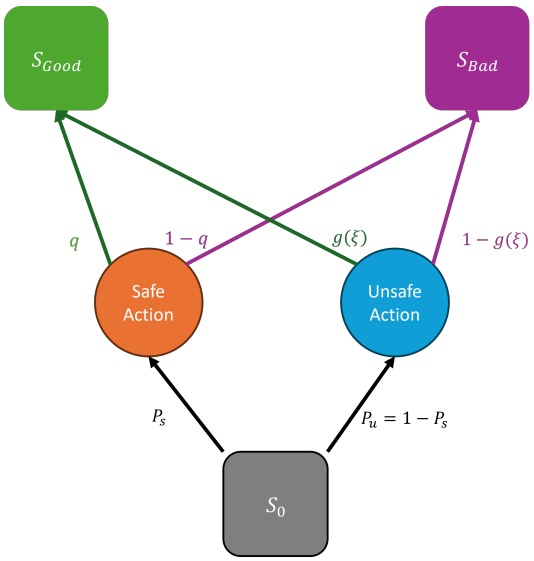


Figure 5: Toy MDP Diagram with varying transition dynamics indexed by the domain parameter $\xi \in [0,1]^d$. The policy and the sampler are updated alternately using return information $J(a, \xi)$.

where $q \in [0,1]$ is a fixed baseline success probability independent of the domain parameter. The unsafe action induces the transition

$$P(S_{\text{good}} \mid S_0, a_u, \xi) = g(\xi), \qquad P(S_{\text{bad}} \mid S_0, a_u, \xi) = 1 - g(\xi),$$

where $g(\xi) \in [0,1]$ is a domain-dependent success probability. Thus, the safe action has fixed and domain-insensitive behavior, whereas the unsafe action is sensitive to the current domain parameter ξ.

Because the episode terminates immediately after this transition, the action values are simply the success probabilities:

$$J(a_s, \xi) = q, \qquad J(a_u, \xi) = g(\xi).$$

Equivalently, q can be viewed as the expected return of the safe action and $g(\xi)$ as the expected return of the unsafe action under domain parameter ξ. This gives the toy MDP a clear interpretation: the unsafe action can outperform the safe action, but only in favorable regions of the domain space.

We parameterize the policy by

$$\pi = (p_s, p_u), \qquad p_s + p_u = 1,$$

where $p_u \in [0,1]$ denotes the probability of selecting the unsafe action at S_0. Although each episode is one-step, the target distribution over domain parameters changes across policy updates through policy-dependent exponential tilting. Given the current policy π_t, the policy induces a tilted target distribution over domain parameters,

$$\nu_\beta(\xi \mid \pi) = \frac{\exp\big(\beta d\, p_u\, g(\xi)\big)}{Z(\lambda)} \mathbf{1}_{[0,1]^d}(\xi) \qquad \lambda := \beta d\, p_u,$$

where

$$Z(\lambda) = \int_{[0,1]^d} \exp\big(\lambda\, g(\xi)\big)\, d\xi.$$

Here β is the risk-sensitivity parameter. The factor d is specific to these synthetic targets, whose scores are normalized to $[0,1]$ as averages over d coordinates; it keeps the tilt per coordinate comparable across dimensions. In the MuJoCo Playground experiments, β is applied directly to the normalized return, and no dimension-dependent scaling is used. Thus, the current policy changes the distribution of domain parameters encountered in future episodes through exponential tilting.

Training episodes are *not* drawn from $\nu_\beta(\cdot|\pi_t)$ directly. As in full RSDR (Algorithm 1), a learned sampler ν_ϕ (GMMVI or DIS-LV; Appendix D) is updated to track $\nu_\beta(\cdot|\pi_t)$, and the domain parameters

used for the policy update are drawn from this learned sampler. We write ϕ_t for the sampler parameters after the sampler update toward $\nu_\beta(\cdot|\pi_t)$.

This is the toy analogue of the sampler update in RSDR: the policy induces a soft adversarial target distribution over domains, and the sampler is trained to track that target.

Under this target distribution, the safe action value remains the constant

$$Q_\beta(a_s) = q,$$

while the unsafe action value is the expectation of the score under the induced domain distribution,

$$Q_\beta(a_u) = \mathbb{E}_{\xi\sim\nu_\beta(\cdot|\pi)}[g(\xi)].$$

The closed-form $Q_\beta(a_u)$ serves only as an analytic reference and is *not* used to update the policy. Instead, the policy uses the Monte Carlo estimate of the unsafe-action value under the learned sampler,

$$\widehat{Q}_t(a_u) = \frac{1}{B}\sum_{i=1}^{B} g(\xi_i;\pi_t), \quad \xi_i \sim \nu_{\phi_t}, \quad B = 4096,$$

and is updated through a soft best response,

$$p_u = \sigma\Big(\tau\big(\widehat{Q}_t(a_u) - q\big)\Big), \qquad \sigma(z) = \frac{1}{1+e^{-z}},$$

where $\tau > 0$ is an inverse-temperature parameter. Altogether, this defines the closed-loop recursion

$$\pi_t \longrightarrow \nu_\beta(\cdot \mid \pi_t) \xrightarrow{\text{sampler update}} \nu_{\phi_t} \longrightarrow \widehat{Q}_t(a_u) \longrightarrow \pi_{t+1}.$$

Because the policy is driven by $\widehat{Q}_t(a_u)$, the trajectory $\{p_{u,t}\}$ inherits the sampler's estimation error $\widehat{Q}_t(a_u) - Q_\beta(a_u)$ and is not expected to coincide with the idealized recursion that uses the exact $Q_\beta(a_u)$; different samplers can therefore drive the policy to different values of p_u. Tracking quality is always measured against the exact target $\nu_\beta(\cdot \mid \pi_t)$ at the policy π_t actually realized in this closed loop (Appendix E.1.2).

This one-step construction isolates the central mechanism of RSDR in the simplest *analytically tractable* setting: the policy induces a target distribution over domain parameters, and the learned sampler must track that target as the policy evolves. The toy example is therefore intended as a mechanistic study of sampler tracking rather than a standalone control benchmark. We use $q = 0.6, \tau = 1$ across whole toy example experiments.

E.1.1 Three target families

Within this common toy-MDP template, we consider three analytically tractable target families of increasing structural complexity: a *Tilted Linear Target*, a *Fixed-Mode Cosine Target*, and a *Policy-Shifted Cosine Target*. The first two are instances of the toy MDP above, with a fixed success probability $g(\xi)$ for the unsafe action. The third is an additional synthetic tracking benchmark in which the score $g(\xi;\pi)$ itself moves with the policy. These respectively test concentration tracking for a single-mode target, concentration tracking around fixed periodic modes, and tracking of a moving multimodal target whose peak locations depend on the policy.

Tilted linear target. For the tilted linear target, we choose a linear score on the unit box,

$$g_{\text{lin}}(\xi;\pi) = \frac{1}{d}\sum_{i=1}^{d} x_i, \qquad \xi = (x_1,\dots,x_d) \in [0,1]^d.$$

The induced distribution is a separable truncated-exponential family,

$$\nu_\lambda(\xi) \propto \exp\left(\frac{\lambda}{d}\sum_{i=1}^{d}\xi_i\right)\mathbf{1}_{[0,1]^d}(\xi), \qquad \frac{\lambda}{d} = \beta p_u. \tag{31}$$

Thus, the policy changes only the tilt strength, not the underlying geometry of the score landscape. For $\lambda > 0$, probability mass shifts toward $\xi^\star = (1,\dots,1)$; for $\lambda < 0$, it shifts toward $\xi^\star = (0,\dots,0)$;

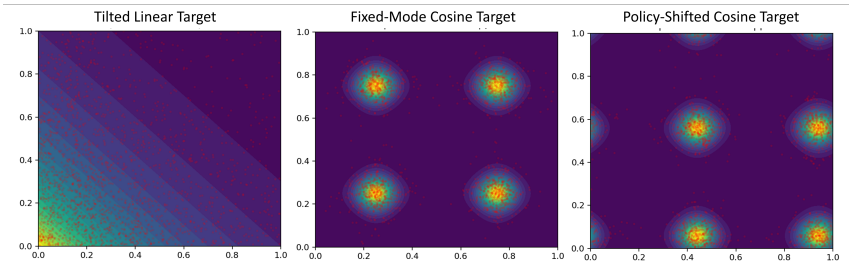


Figure 6: Toy target distributions over $\xi \in [0,1]^2$: tilted linear (left), fixed-mode cosine (middle), and policy-shifted cosine (right). The heatmap shows the policy-induced tilted target, and red dots show samples from the learned GMMVI sampler. These examples illustrate increasing difficulty in sampler tracking, culminating in a moving multimodal target whose modes shift with the policy.

and for $\lambda = 0$, the target reduces to the uniform distribution. This gives the simplest single-mode sanity-check setting, in which the sampler only needs to track changes in concentration. The corresponding unsafe-action value is the truncated-exponential mean

$$Q_\beta(a_u) = \frac{e^{\beta p_u}}{e^{\beta p_u} - 1} - \frac{1}{\beta p_u},$$

with continuous extension $Q_\beta(a_u) = 1/2$ at $\beta p_u = 0$. We use $\beta = -5$ for this target.

Fixed-mode cosine target. For the fixed-mode cosine target, we replace the linear score with a separable cosine landscape,

$$g_{\text{fix}}(\xi) = c + \sum_{i=1}^{d} a_i \cos(2\pi k_i x_i + \phi_i), \tag{32}$$

with parameters chosen so that $g_{\text{fix}}(\xi) \in [0,1]$. The resulting exponentially tilted family remains analytically tractable and has exact normalizer

$$Z(\lambda) = e^{\lambda c} \prod_{i=1}^{d} I_0(\lambda a_i),$$

where $I_0(\cdot)$ is the modified Bessel function of the first kind of order 0. Here the policy still enters only through the scalar tilt $\lambda = \beta d\, p_u$. Consequently, the mode locations are fixed: since $\beta < 0$ and $p_u > 0$, we have $\lambda < 0$ throughout, and the modes stay at the minima of g_{fix}. The policy instead controls how sharply the mass concentrates around each mode, through the von Mises concentration $\kappa_i = |\lambda a_i|$. This isolates the problem of tracking a changing multimodal distribution over a fixed geometric landscape. The unsafe-action value is

$$Q_\beta(a_u) = c + \sum_{i=1}^{d} a_i \frac{I_1(\lambda a_i)}{I_0(\lambda a_i)}, \qquad \lambda = \beta d\, p_u.$$

We fix for $\forall i \in \{1, \cdots, d\}$, $k_i = 1$ or 2, $a_i = 0.35/d$, $\phi_i = 0.1$ and $c = 0.5$, $\beta = -10$ for experiments. To avoid excessive multimodality, we additionally constrain the total frequency by requiring $\sum_{i=1}^{d} k_i \leq 100$.

Policy-shifted cosine target. For policy-shifted cosine target, we make the cosine centers policy-dependent:

$$g_{\text{shift}}(\xi; \pi) = c + \sum_{i=1}^{d} a_i \cos\big(2\pi k_i (x_i - \mu_i(\pi))\big), \qquad \mu_i(\pi) = \mu_{i,0} + b_i p_u \pmod 1. \tag{33}$$

Now the current policy changes both the tilt strength and the mode locations. The success probability of the unsafe action, $g_{\text{shift}}(\xi; \pi)$, now depends on the policy, so the environment itself changes with

the policy, not only the domain distribution. The target therefore becomes not only multimodal but also spatially moving. Importantly, exact normalization is still preserved,

$$Z(\lambda, \pi) = e^{\lambda c} \prod_{i=1}^{d} I_0(\lambda a_i),$$

so the family remains analytically simple even though its geometry changes with the policy. The unsafe-action value retains the same closed form as in the fixed-mode case,

$$Q_\beta(a_u) = c + \sum_{i=1}^{d} a_i \frac{I_1(\lambda a_i)}{I_0(\lambda a_i)}, \qquad \lambda = \beta d\, p_u,$$

while the peak locations move according to $\mu_i(\pi)$.

This is the clearest toy analogue of the tracking problem addressed by RSDR: the sampler must follow a policy-induced tilted target whose peaks move as the policy evolves. We use same configuration with fixed version, additionally set $\mu_{i,0} = 0, b_i \in [-2, 2]$ for experiment.

Taken together, these three target families provide controlled tests of increasing complexity for sampler tracking. The Tilted Linear Target tests concentration tracking for a single-mode target, the Fixed-Mode Cosine Target tests changing concentration around fixed periodic modes, and the additional Policy-Shifted Cosine benchmark tests simultaneous changes in concentration and mode locations. Their common purpose is to illustrate tracking of an evolving policy-induced target distribution in an analytically tractable setting.

E.1.2 Evaluation results and metrics

A useful feature of the toy setting is that we can sample directly from the policy-induced target distribution, which enables quantitative evaluation of sampler-tracking quality. For the truncated-exponential target in Eq. (31), target samples are obtained by inverse-transform sampling. For the cosine-based targets in Eqs. (32)–(33), target samples are obtained from the corresponding von Mises-type construction in Best and Fisher [1979].

Because we have direct access to samples from the target distribution, we can measure the *Sinkhorn distance* [Cuturi, 2013] and an energy based *Wasserstein distance* [Panaretos and Zemel, 2019] between learned and target samples. Two metrics measure how accurately the learned sampler tracks the current policy-induced tilted target. For the Wasserstein-type metric, we follow the energy-space evaluation protocol used in prior sampling work in Liu et al. [2025]; for the Sinkhorn distance, we follow the implementation of Blessing et al. [2024]. For the energy-based metric, we evaluate the target energy $E(\xi) = -\lambda\, g(\xi; \pi_t)$, i.e., the negative log-density of $\nu_\beta(\cdot \mid \pi_t)$ up to an additive constant. Energies are evaluated in the original box coordinates, and the reported value is the squared 2-Wasserstein distance between the two empirical energy distributions, i.e., the mean of squared differences of sorted energies, as in Liu et al. [2025].

We compare the default reverse-KL sampler (GMMVI) against the alternative DIS-LV sampler (Appendix D.2) with those two metrics. Table 3 reports initial-to-final toy-MDP tracking metrics for GMMVI and DIS-LV. Overall, the task becomes more difficult from the Tilted Linear target to the Policy-Shifted Cosine target and as dimensionality increases. GMMVI is usually stronger on the Tilted Linear and Fixed-Mode Cosine targets, while the Policy-Shifted Cosine target is more mixed: DIS-LV is competitive and sometimes better on the energy-based metric, whereas GMMVI is often better on Sinkhorn distance in higher dimensions.

Across all three settings, the distribution-matching curves quantify how well the sampler follows the current policy-induced target rather than fitting a static distribution once and for all. This supports the interpretation of RSDR as a sampler-tracking method over a policy-dependent tilted distribution, and provides a controlled comparison between GMMVI and DIS-LV in a setting where the ground-truth target distribution is analytically available.

E.2 Comparison with domain randomization baselines

We present a comprehensive performance evaluation—encompassing mean return, CVaR_{20}, CVaR_{10}, and minimum return—for all evaluated domain randomization algorithms in Table 4 and Table 5. These tables also illustrate the sensitivity of both average and left-tail metrics to the risk parameter β.

Table 3: Toy-MDP tracking results across target families and latent dimensions. Each entry reports the initial values $\rightarrow$ final values from the metric logs. Lower Sinkhorn distance and lower Energy $\mathcal{W}_2^2$ indicate better agreement with the analytically sampled target distribution.

Target	d	Method	p_u	Sinkhorn	Energy $\mathcal{W}_2^2$
Tilted Linear ($\beta = -5$)	2	GMMVI	$0.872 \rightarrow 0.461$	$0.02 \rightarrow 0.00137$	$0.024 \rightarrow 4.1 \times 10^{-5}$
		DIS-LV	$0.866 \rightarrow 0.367$	$0.131 \rightarrow 0.00308$	$0.209 \rightarrow 1.1 \times 10^{-4}$
	5	GMMVI	$0.872 \rightarrow 0.461$	$0.093 \rightarrow 0.021$	$0.017 \rightarrow 2.4 \times 10^{-5}$
		DIS-LV	$0.866 \rightarrow 0.367$	$0.34 \rightarrow 0.029$	$0.185 \rightarrow 3.6 \times 10^{-5}$
	10	GMMVI	$0.872 \rightarrow 0.461$	$0.302 \rightarrow 0.185$	$0.018 \rightarrow 1.2 \times 10^{-5}$
		DIS-LV	$0.865 \rightarrow 0.367$	$0.757 \rightarrow 0.223$	$0.178 \rightarrow 5.1 \times 10^{-5}$
	20	GMMVI	$0.872 \rightarrow 0.461$	$0.944 \rightarrow 0.75$	$0.021 \rightarrow 3.5 \times 10^{-5}$
		DIS-LV	$0.865 \rightarrow 0.367$	$1.742 \rightarrow 0.873$	$0.18 \rightarrow 3.8 \times 10^{-5}$
Fixed-Mode Cosine ($\beta = -10$)	2	GMMVI	$0.869 \rightarrow 0.429$	$0.00567 \rightarrow 0.00154$	$3.096 \rightarrow 0.00131$
		DIS-LV	$0.872 \rightarrow 0.454$	$0.041 \rightarrow 0.0065$	$21.373 \rightarrow 0.925$
	5	GMMVI	$0.869 \rightarrow 0.424$	$0.039 \rightarrow 0.019$	$22.56 \rightarrow 0.00813$
		DIS-LV	$0.872 \rightarrow 0.454$	$0.118 \rightarrow 0.036$	$142.349 \rightarrow 8.074$
	10	GMMVI	$0.868 \rightarrow 0.412$	$0.177 \rightarrow 0.159$	$120.707 \rightarrow 0.093$
		DIS-LV	$0.872 \rightarrow 0.453$	$0.325 \rightarrow 0.197$	$694.851 \rightarrow 68.191$
	20	GMMVI	$0.868 \rightarrow 0.406$	$0.958 \rightarrow 0.936$	$553.039 \rightarrow 0.028$
		DIS-LV	$0.872 \rightarrow 0.453$	$1.297 \rightarrow 0.969$	$3 \times 10^3 \rightarrow 368.957$
Policy-Shifted Cosine ($\beta = -10$)	2	GMMVI	$0.869 \rightarrow 0.429$	$0.076 \rightarrow 0.061$	$4.293 \rightarrow 3.959$
		DIS-LV	$0.87 \rightarrow 0.453$	$0.127 \rightarrow 0.053$	$3.457 \rightarrow 1.048$
	5	GMMVI	$0.869 \rightarrow 0.427$	$0.131 \rightarrow 0.15$	$20.608 \rightarrow 4.864$
		DIS-LV	$0.87 \rightarrow 0.449$	$0.285 \rightarrow 0.188$	$15.875 \rightarrow 1.805$
	10	GMMVI	$0.868 \rightarrow 0.412$	$0.708 \rightarrow 0.488$	$314.11 \rightarrow 170.887$
		DIS-LV	$0.871 \rightarrow 0.429$	$1.207 \rightarrow 0.786$	$138.748 \rightarrow 185.403$
	20	GMMVI	$0.868 \rightarrow 0.406$	$1.385 \rightarrow 1.106$	$937.393 \rightarrow 249.86$
		DIS-LV	$0.87 \rightarrow 0.42$	$1.803 \rightarrow 1.592$	$406.322 \rightarrow 205.539$

Across the six tasks and $\beta \in \{-10, -20, -30\}$, all 18 CVaR_{10} point estimates and 15 of 18 minimum-return point estimates exceed UDR. A common choice of $\beta = -20$ improves CVaR_{10} by 6.7%–24.0% over UDR across all six tasks, with mean-return changes ranging from -2.3% to $+5.1\%$. We therefore regard $\beta = -20$ as a practical default within our episode-length-normalized return setup, while the preferred setting remains task- and metric-dependent.

Because the exponential tilt reweights dynamics parameters according to their expected returns, the average return can exhibit a slight degradation as β moves further from the risk-neutral region ($\beta = 0$). This observation aligns with the inherent trade-offs in minimax optimization, where robustness to worst-case scenarios typically comes at the cost of average-case optimality.

However, in the context of zero-shot sim-to-real deployment via domain randomization, prioritizing left-tail robustness is highly advantageous. This approach aligns with the core philosophy of Robust RL [Padakandla, 2021], which posits that ensuring reliability under adversarial or outlier dynamics is more critical for real-world stability than maximizing performance under idealized average conditions.

The training progress across various risk-sensitivity regimes is depicted in Figure 7, illustrating the distinct learning behaviors associated with each choice of β.

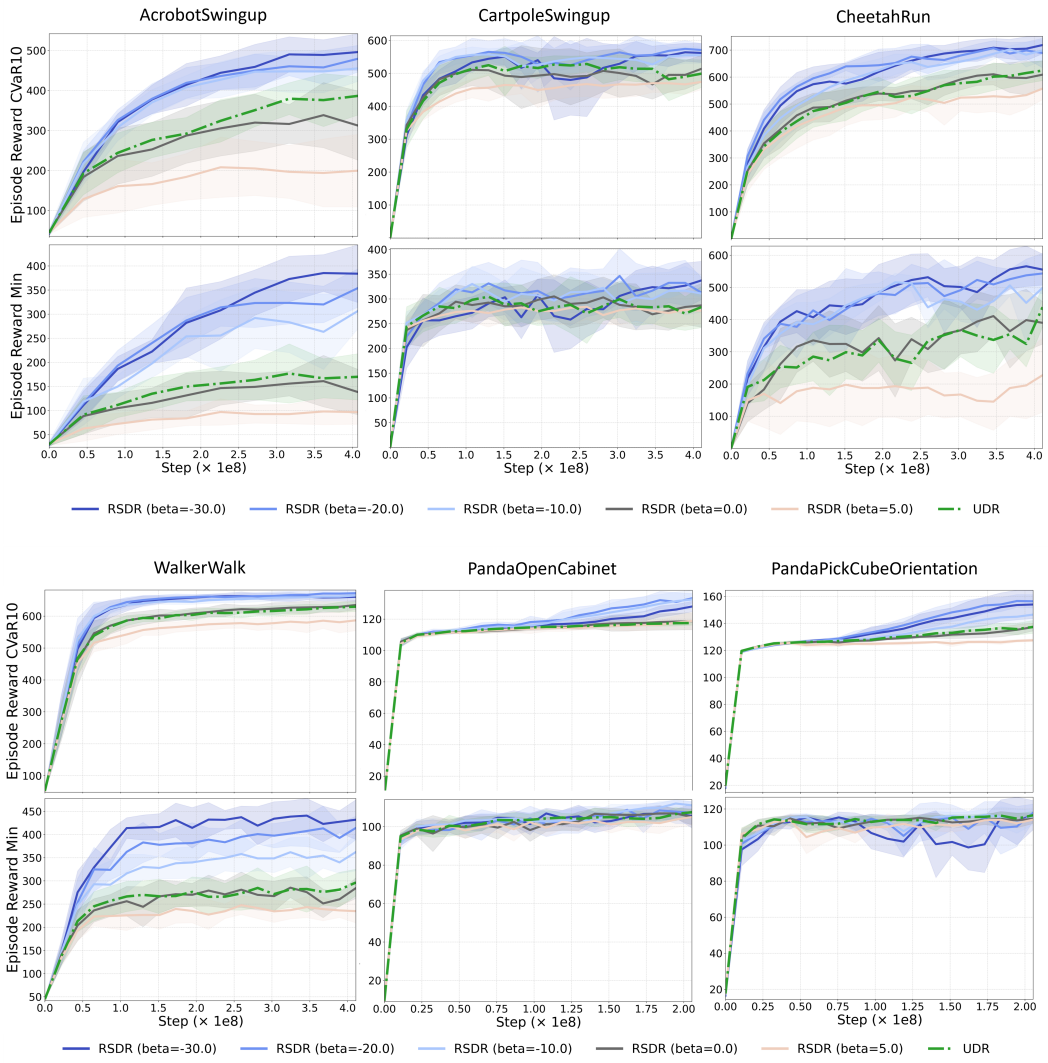


Figure 7: Performance of RSDR with varying values of β. The top row illustrates the CVaR_{10} return, while the bottom row shows the minimum of the episodic return. Risk-averse ($\beta < 0$) RSDR shows stronger robustness than Risk-Neutral ($\beta = 0$ or UDR) and Risk-Seeking ($\beta > 0$) RSDR.

Table 4: Domain randomization results across MuJoCo Playground benchmarks (DM Control Suite)

(a) AcrobotSwingup

DR algorithm	**Mean**	**CVaR$_{20}$**	**CVaR$_{10}$**	**Min**
RSDR [$\beta = -30$]	**579.49 ± 36.11**	**514.23 ± 43.85**	**496.01 ± 45.81**	**383.93 ± 59.77**
RSDR [$\beta = -20$]	559.61 ± 31.91	498.42 ± 32.61	479.26 ± 33.39	354.05 ± 39.80
RSDR [$\beta = -10$]	545.34 ± 34.92	477.34 ± 34.04	455.12 ± 32.51	306.50 ± 37.12
RSDR [$\beta = 0$]	486.95 ± 70.06	355.90 ± 93.62	312.35 ± 86.54	138.46 ± 47.14
RSDR [$\beta = 5$]	471.87 ± 57.81	243.59 ± 99.58	199.42 ± 90.68	96.72 ± 25.22
AutoDR	272.77 ± 45.98	78.68 ± 18.25	67.36 ± 15.41	38.34 ± 8.24
DORAEMON	315.56 ± 36.42	95.87 ± 36.74	79.54 ± 29.46	43.56 ± 12.90
GoFlow	429.59 ± 38.27	304.87 ± 21.39	265.38 ± 26.19	118.98 ± 36.27
UDR	532.27 ± 30.52	432.32 ± 38.84	386.42 ± 48.05	170.02 ± 47.82

(b) CartpoleSwingup

DR algorithm	**Mean**	**CVaR$_{20}$**	**CVaR$_{10}$**	**Min**
RSDR [$\beta = -30$]	729.40 ± 15.92	600.76 ± 27.72	561.94 ± 28.94	**337.47 ± 38.53**
RSDR [$\beta = -20$]	745.43 ± 12.70	**614.34 ± 17.48**	**570.84 ± 19.43**	312.58 ± 30.24
RSDR [$\beta = -10$]	755.93 ± 5.90	609.24 ± 16.45	555.68 ± 24.86	310.81 ± 37.43
RSDR [$\beta = 0$]	757.54 ± 10.71	585.03 ± 26.16	513.74 ± 40.93	286.97 ± 44.19
RSDR [$\beta = 5$]	736.94 ± 12.92	548.53 ± 19.14	473.87 ± 23.54	276.19 ± 25.92
AutoDR	736.35 ± 26.34	558.64 ± 50.25	490.32 ± 64.80	264.58 ± 61.69
DORAEMON	711.79 ± 53.86	510.33 ± 117.57	444.98 ± 122.99	275.37 ± 59.07
GoFlow	718.20 ± 60.66	540.83 ± 84.58	470.83 ± 82.87	277.24 ± 44.19
UDR	746.65 ± 7.62	570.62 ± 25.25	499.04 ± 41.93	283.61 ± 41.58

(c) CheetahRun

DR algorithm	**Mean**	**CVaR$_{20}$**	**CVaR$_{10}$**	**Min**
RSDR [$\beta = -30$]	846.02 ± 22.66	**750.83 ± 30.74**	**722.14 ± 32.45**	**563.43 ± 52.16**
RSDR [$\beta = -20$]	848.40 ± 49.12	749.61 ± 54.24	714.02 ± 52.11	546.08 ± 64.00
RSDR [$\beta = -10$]	868.03 ± 20.01	747.90 ± 34.97	700.95 ± 35.29	503.69 ± 59.58
RSDR [$\beta = 0$]	856.25 ± 24.89	683.59 ± 48.48	626.40 ± 45.37	410.39 ± 71.85
RSDR [$\beta = 5$]	847.21 ± 13.03	630.01 ± 33.97	557.23 ± 40.92	227.48 ± 115.30
AutoDR	858.77 ± 15.70	695.53 ± 37.08	629.88 ± 50.13	378.58 ± 112.91
DORAEMON	863.98 ± 14.16	689.03 ± 41.98	625.02 ± 39.75	290.79 ± 110.51
GoFlow	861.91 ± 16.24	694.06 ± 34.91	634.83 ± 34.15	381.66 ± 80.79
UDR	866.24 ± 12.54	703.71 ± 33.65	641.75 ± 33.99	446.00 ± 71.00

(d) WalkerWalk

DR algorithm	**Mean**	**CVaR$_{20}$**	**CVaR$_{10}$**	**Min**
RSDR [$\beta = -30$]	794.82 ± 12.46	695.65 ± 15.24	660.59 ± 14.53	**432.03 ± 42.81**
RSDR [$\beta = -20$]	837.52 ± 6.34	721.47 ± 9.68	**671.70 ± 10.07**	414.28 ± 25.32
RSDR [$\beta = -10$]	853.59 ± 3.52	**723.69 ± 7.86**	664.50 ± 9.36	362.36 ± 40.63
RSDR [$\beta = 0$]	857.89 ± 5.41	707.90 ± 11.72	634.76 ± 14.33	284.45 ± 22.31
RSDR [$\beta = 5$]	856.31 ± 5.64	678.30 ± 22.41	586.70 ± 32.29	234.76 ± 25.35
AutoDR	848.67 ± 5.60	663.74 ± 29.37	569.02 ± 42.90	224.66 ± 26.06
DORAEMON	857.27 ± 4.15	704.40 ± 8.69	629.17 ± 12.47	280.08 ± 33.67
GoFlow	855.46 ± 2.26	703.57 ± 13.83	629.23 ± 20.21	286.46 ± 39.39
UDR	856.91 ± 4.66	703.56 ± 11.82	629.43 ± 15.26	296.47 ± 28.78

Table 5: Domain randomization results across MuJoCo Playground benchmarks (Manipulation)

(a) PandaOpenCabinet

DR algorithm	**Mean**	**CVaR$_{20}$**	**CVaR$_{10}$**	**Min**
RSDR [$\beta = -30$]	173.53 ± 5.83	135.84 ± 4.13	128.03 ± 3.29	105.54 ± 6.22
RSDR [$\beta = -20$]	177.51 ± 1.59	**142.08 ± 2.88**	**133.30 ± 2.99**	107.28 ± 5.78
RSDR [$\beta = -10$]	179.16 ± 4.10	140.91 ± 6.31	132.47 ± 5.90	**111.10 ± 1.95**
RSDR [$\beta = 0$]	171.13 ± 4.90	123.56 ± 7.68	119.02 ± 5.71	105.10 ± 4.58
RSDR [$\beta = 5$]	169.93 ± 6.06	123.27 ± 7.99	118.83 ± 5.45	105.01 ± 4.10
AutoDR	121.66 ± 15.09	97.44 ± 4.10	95.07 ± 5.06	83.33 ± 14.82
DORAEMON	123.79 ± 14.45	98.39 ± 1.17	96.96 ± 1.24	87.26 ± 5.59
GoFlow	138.18 ± 31.41	119.04 ± 11.19	114.88 ± 7.61	99.10 ± 3.82
UDR	171.63 ± 3.29	122.22 ± 2.56	117.48 ± 1.29	107.53 ± 1.32

(b) PandaPickCubeOrientation

DR algorithm	**Mean**	**CVaR$_{20}$**	**CVaR$_{10}$**	**Min**
RSDR [$\beta = -30$]	209.15 ± 9.77	166.81 ± 11.49	154.00 ± 10.42	116.60 ± 8.11
RSDR [$\beta = -20$]	218.38 ± 2.63	**170.90 ± 5.46**	**156.38 ± 4.91**	**117.91 ± 4.77**
RSDR [$\beta = -10$]	220.06 ± 3.11	163.01 ± 6.41	146.49 ± 5.74	115.34 ± 4.11
RSDR [$\beta = 0$]	218.97 ± 2.73	152.26 ± 3.14	137.23 ± 2.85	114.59 ± 2.71
RSDR [$\beta = 5$]	214.50 ± 3.18	137.33 ± 3.01	127.45 ± 1.05	114.10 ± 1.82
AutoDR	200.15 ± 3.70	119.31 ± 4.81	112.08 ± 3.72	89.79 ± 6.07
DORAEMON	201.92 ± 2.90	121.29 ± 2.86	113.82 ± 2.44	85.94 ± 11.55
GoFlow	207.91 ± 3.70	133.31 ± 4.12	127.24 ± 2.45	114.51 ± 3.05
UDR	219.09 ± 4.85	151.56 ± 7.46	137.35 ± 4.83	116.52 ± 2.06

E.2.1 2D RSDR Results with heatmap visualization

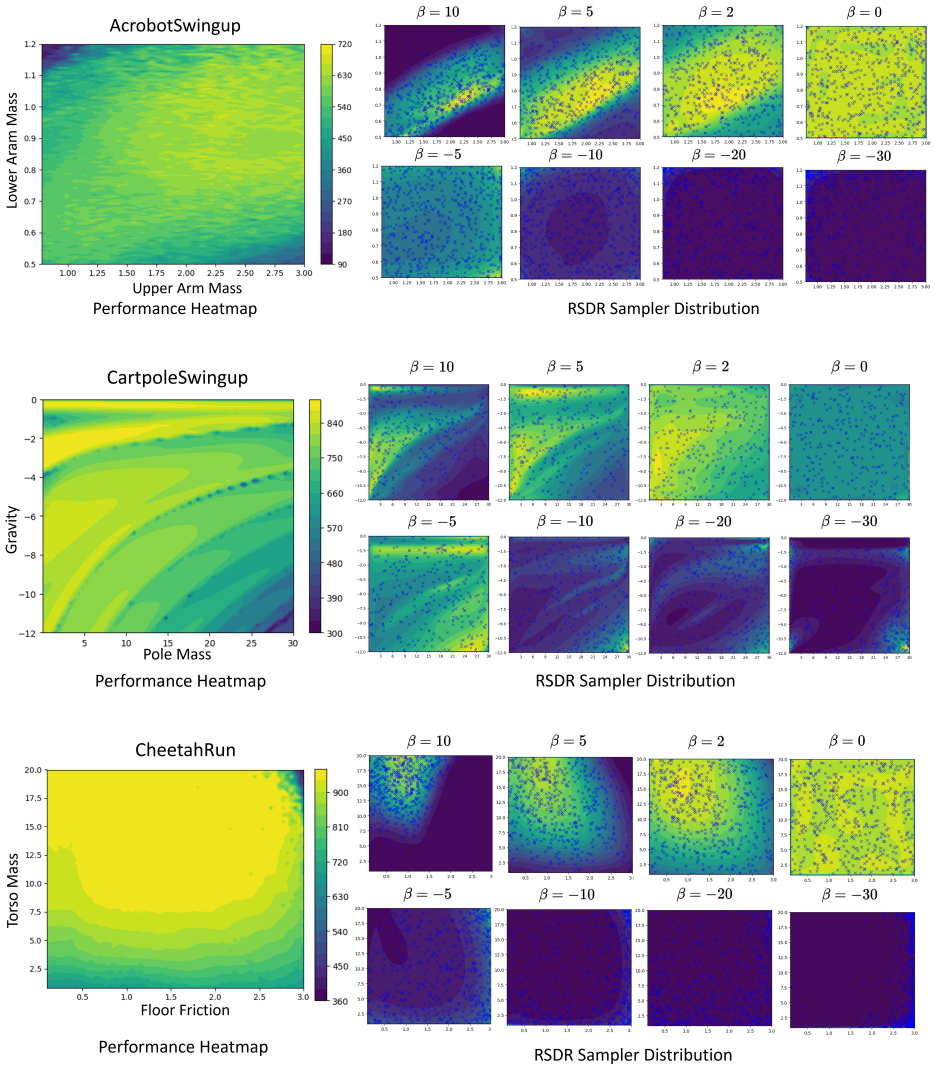


Figure 8: Visualizing the learned sampler ν_ϕ targeting $\nu_\beta \propto \exp(\beta J(\theta; \xi))$ for AcrobotSwingup(top), CartpoleSwingup(middle), and CheetahRun (bottom). RSDR successfully transitions from risk-averse ($\beta < 0$) to optimistic ($\beta > 0$) regimes, and approximates the uniform UDR distribution at $\beta = 0$.

To investigate the relationship between the risk parameter β and the resulting sampling distributions, we visualize the learned samplers ν_ϕ on the AcrobotSwingup, CartpoleSwingup and CheetahRun tasks. Figure 8 qualitatively illustrates the intended return-dependent reweighting across the illustrated risk regimes:

- **Optimistic regime ($\beta > 0$):** The sampler concentrates mass on the "easy" regions of the parameter space Ξ where the current policy π_θ achieves maximum reward. This effectively generates an automated curriculum that emphasizes regions of high performance.
- **Risk-averse regime ($\beta < 0$):** Conversely, the sampler shifts toward "adversarial" regions where the policy underperforms. By focusing training on these "tail-end" failure modes, RSDR enhances the policy's robustness to worst-case scenarios.
- **Neutral regime ($\beta = 0$):** The target reduces to the uniform reference ν_0 used by UDR, and the learned sampler approximates it, providing a neutral coverage of the entire parameter space.

Table 6: Sampler-tracking diagnostics throughout policy training. TV and JS compare the learned sampler to the instantaneous exponential-tilting target on a 128×128 grid. Values are mean $\pm$ standard deviation over three seeds.

Task	Metric	0%	25%	50%	75%	100%
AcrobotSwingup	TV	0.870 ± 0.006	0.165 ± 0.033	0.123 ± 0.029	0.101 ± 0.019	0.143 ± 0.023
	JS	0.510 ± 0.006	0.021 ± 0.001	0.012 ± 0.001	0.018 ± 0.001	0.017 ± 0.007
CheetahRun	TV	0.870 ± 0.006	0.259 ± 0.089	0.103 ± 0.019	0.334 ± 0.071	0.218 ± 0.053
	JS	0.520 ± 0.006	0.061 ± 0.021	0.011 ± 0.003	0.090 ± 0.028	0.039 ± 0.014

The heatmaps demonstrate that RSDR maintains high expressivity, capturing the non-convex contours of the episode reward landscape $J(\theta; \xi)$ without mode collapse.

We also performed time-resolved diagnostics on two representative 2-D benchmarks, where a grid-discretized target can be constructed without introducing a separate high-dimensional density estimator. At 0/25/50/75/100% of training, we evaluate the current policy over a 128×128 grid (16,384 configurations; finer than the 64×64 evaluation grid of Appendix F), form the normalized-return target $\nu_\beta(\xi) \propto \exp(\beta \tilde{J}(\theta; \xi))\, \nu_0(\xi)$ (with uniform ν_0 in these tasks), and compare it with the learned sampler on the same grid ($\beta = -10$, three seeds). We report total variation (TV) and Jensen–Shannon (JS) divergence, which are bounded and less sensitive to tiny grid probabilities than directional KL estimates in Table 6.

The distances drop sharply after initialization and remain much smaller throughout training. Acrobot shows consistently close tracking; Cheetah exhibits a temporary lag at 75% followed by partial re-tracking at 100%. These results support the intended moving-target mechanism while also showing that tracking is approximate and can lag as the policy changes.

E.2.2 Adaptive scheduling with realized radius feedback control

The main experiments use a fixed temperature, corresponding to a KL-penalized entropic-risk objective rather than directly enforcing a prescribed KL radius. By contrast, the prescribed-radius objective additionally optimizes the dual temperature:

$$J_{\mathrm{DRO},c}(\theta) = \sup_{\beta<0} \left\{ \mathcal{R}_\beta(\theta) + \frac{c}{\beta} \right\}.$$

Thus, a fixed β does not generally solve the KL-DRO problem for a prescribed radius c. Instead, it induces a policy-dependent effective radius $c_\theta(\beta) = D_{\mathrm{KL}}(\nu_{\beta,\theta} \| \nu_0)$. To investigate adaptive temperature selection, we evaluate a proof-of-concept *realized-radius controller*:

$$\beta_{t+1} = \min\Big\{0,\ \beta_t + \eta_\beta \big(\widehat{D}_{\mathrm{KL}}(\nu_{\phi_t} \,\|\, \nu_0) - c\big)\Big\}, \quad \eta_\beta = 1, \beta_0 = -5$$

This is an approximate feedback rule rather than an exact constrained solver, as it regulates the learned sampler's realized KL while the policy-dependent target distribution evolves.

Performance is reported as mean and standard deviation over 10 seeds, and realized KL and late β are averaged over the final 10% of training.

As shown in Table 7, the controller approximately tracks the prescribed KL budgets. The same $c = 0.5$ yields nearly identical realized KL on the two tasks but very different late temperatures, directly illustrating the task-dependent $c \leftrightarrow \beta$ mapping. On WalkerWalk, larger KL budgets can improve tail performance at the expense of mean return. On AcrobotSwingup, both $c = 0.5$ and $c = 1.0$ improve all three metrics over UDR (Mean/CVaR$_{10}$/Minimum: 532.27/386.42/170.02). A tuned fixed temperature remains stronger on some tail metrics. These results demonstrate the interpretability and approximate calibration of the adaptive controller, rather than its uniform superiority over fixed temperature selection.

E.3 Comparison with EPOpt (robust RL baseline)

We additionally compare RSDR against EPOpt [Rajeswaran et al., 2016], a classical robust reinforcement learning baseline that optimizes the policy using only the worst-performing fraction of sampled

Table 7: Adaptive-temperature RSDR compared with fixed-temperature RSDR. Returns are mean $\pm$ standard deviation over 10 seeds. Realized KL and late β are averaged over the final 10% of training. Bold indicates the highest mean, CVaR_{10}, or minimum return within each task.

(a) AcrobotSwingup

Method	**Realized KL**	**Late β**	**Mean**	**CVaR_{10}**	**Min**
Adaptive-β ($c = 0.5$)	0.522 ± 0.14	-38.33 ± 5.09	559.79 ± 52.59	474.12 ± 54.13	336.51 ± 39.70
Adaptive-β ($c = 1.0$)	0.84 ± 0.21	-64.17 ± 5.16	536.70 ± 37.17	445.76 ± 40.69	314.73 ± 41.82
Fixed $\beta = -30$	0.41 ± 0.23	-30 (fixed)	$\mathbf{579.49 \pm 36.11}$	$\mathbf{496.01 \pm 45.81}$	$\mathbf{383.93 \pm 59.77}$

(b) WalkerWalk

Method	**Realized KL**	**Late β**	**Mean**	**CVaR_{10}**	**Min**
Adaptive-β ($c = 0.5$)	0.516 ± 0.14	-8.17 ± 0.48	851.87 ± 5.12	655.01 ± 8.14	322.19 ± 28.10
Adaptive-β ($c = 1.0$)	0.95 ± 0.02	-12.01 ± 0.57	$\mathbf{852.48 \pm 2.37}$	666.87 ± 6.00	347.79 ± 50.72
Adaptive-β ($c = 2.0$)	1.91 ± 0.20	-18.41 ± 1.46	839.80 ± 5.44	669.68 ± 8.72	380.42 ± 25.29
Fixed $\beta = -20$	2.12 ± 0.27	-20 (fixed)	837.52 ± 6.34	$\mathbf{671.70 \pm 10.07}$	414.28 ± 25.32
Fixed $\beta = -30$	3.30 ± 0.21	-30 (fixed)	794.82 ± 12.46	660.59 ± 14.53	$\mathbf{432.03 \pm 42.81}$

trajectories during training. Concretely, at each policy update, EPOpt selects the bottom ϵ-quantile of trajectories ranked by return and performs the policy update using only those samples. Smaller values of ϵ therefore emphasize a more pessimistic training signal, while larger values approach standard domain-randomized training by incorporating a larger portion of the sampled domains.

To see the effectiveness of EPOpt as a method for selecting sampled dynamics parameters without confounding PPO learning, we compare policies trained with the same learning batch size B. Under this comparison, EPOpt requires an additional $B(\frac{1}{\epsilon} - 1)$ rollouts, whereas UDR and RSDR use the same B rollouts directly for learning.

Table 8 shows that while EPOpt can improve tail robustness relative to UDR for suitable choices of ϵ, its performance is highly sensitive to this quantile hyperparameter. In contrast, risk-averse RSDR achieves stronger and more consistent performance on the tail-focused metrics. These results indicate that directly adapting the training domain distribution through risk-sensitive reweighting is more effective than restricting policy updates to the worst-performing trajectory subset alone.

We also measured end-to-end wall-clock time on the hardware reported in Appendix F using identical PPO hyperparameters and matched effective PPO-update batch size. Results are reported as mean $\pm$ std over three seeds. RSDR adds one sampler-update stage per PPO iteration, corresponding to 512 policy optimizer steps per sampler update in our configuration. Since sampler learning reuses the dynamics and return data collected for PPO, it requires no additional environment interactions. In contrast, EPOpt requires approximately B/ϵ trajectories to retain B trajectories from the lowest-return ϵ fraction under the matched-batch setting.

As shown in Table 9, RSDR incurs less than 2% additional end-to-end training time relative to UDR on both profiled tasks, compared with 5.50%–40.38% for EPOpt.

E.4 Evaluation under non-stationary attack

In this section, we evaluate robustness under non-stationary environmental perturbations, where the dynamics parameters evolve continuously during an episode rather than remaining fixed after reset. This setting is intended to better reflect real-world deployment scenarios in which environmental conditions drift over time [Padakandla, 2021].

As a first non-stationary process, we consider a projected Gaussian random walk driven by zero-mean Gaussian increments in the dynamics parameters. At each time step, the parameters are perturbed by isotropic Gaussian noise and projected back onto the valid domain Ξ:

$$\xi_{t+1} = \text{Proj}_{\Xi}(\xi_t + \sigma\epsilon_t), \qquad \epsilon_t \sim \mathcal{N}(0, \mathbf{I}), \tag{34}$$

where σ controls the perturbation scale and Proj_{Ξ} enforces physical validity of the parameters.

Table 8: Detailed comparison of CVaR_{10} and minimum return against EPOpt across MuJoCo tasks

(a) AcrobotSwingup

Algorithm	CVaR_{10}	Min
EPOpt ($\epsilon = 0.7$)	60.51 ± 24.15	40.21 ± 14.71
EPOpt ($\epsilon = 0.8$)	78.73 ± 46.71	50.52 ± 28.83
EPOpt ($\epsilon = 0.9$)	*360.84 ± 88.85*	*178.76 ± 57.95*
UDR	386.42 ± 48.05	170.02 ± 47.82
RSDR ($\beta = -30$)	$\mathbf{496.01 \pm 45.81}$	$\mathbf{383.93 \pm 59.77}$

(b) CartpoleSwingup

Algorithm	CVaR_{10}	Min
EPOpt ($\epsilon = 0.3$)	0.06 ± 0.08	0.05 ± 0.07
EPOpt ($\epsilon = 0.4$)	99.24 ± 107.55	76.22 ± 94.71
EPOpt ($\epsilon = 0.5$)	106.04 ± 145.64	87.90 ± 120.33
EPOpt ($\epsilon = 0.6$)	302.75 ± 170.41	222.40 ± 125.31
EPOpt ($\epsilon = 0.7$)	510.62 ± 74.13	278.60 ± 75.79
EPOpt ($\epsilon = 0.8$)	*548.44 ± 26.35*	*304.59 ± 15.96*
EPOpt ($\epsilon = 0.9$)	512.05 ± 55.95	293.07 ± 32.26
UDR	499.04 ± 41.93	283.61 ± 41.58
RSDR ($\beta = -20$)	$\mathbf{570.84 \pm 19.43}$	312.58 ± 30.24
RSDR ($\beta = -30$)	561.94 ± 28.94	$\mathbf{337.47 \pm 38.53}$

(c) CheetahRun

Algorithm	CVaR_{10}	Min
EPOpt ($\epsilon = 0.1$)	322.28 ± 194.41	247.72 ± 163.85
EPOpt ($\epsilon = 0.2$)	553.60 ± 201.60	405.61 ± 177.45
EPOpt ($\epsilon = 0.3$)	611.56 ± 137.34	404.78 ± 131.01
EPOpt ($\epsilon = 0.4$)	*697.49 ± 62.73*	*484.97 ± 77.23*
EPOpt ($\epsilon = 0.5$)	676.68 ± 44.88	429.56 ± 70.71
EPOpt ($\epsilon = 0.6$)	653.58 ± 77.42	420.80 ± 49.12
EPOpt ($\epsilon = 0.7$)	694.63 ± 40.36	465.81 ± 47.98
EPOpt ($\epsilon = 0.8$)	663.46 ± 63.12	405.73 ± 55.48
EPOpt ($\epsilon = 0.9$)	622.06 ± 70.83	384.73 ± 70.32
UDR	641.75 ± 33.99	446.00 ± 71.00
RSDR ($\beta = -30$)	$\mathbf{722.14 \pm 32.45}$	$\mathbf{563.43 \pm 52.16}$

(d) WalkerWalk

Algorithm	CVaR_{10}	Min
EPOpt ($\epsilon = 0.2$)	163.30 ± 24.76	108.24 ± 16.92
EPOpt ($\epsilon = 0.3$)	243.65 ± 28.70	179.19 ± 21.48
EPOpt ($\epsilon = 0.4$)	311.38 ± 103.82	186.88 ± 14.61
EPOpt ($\epsilon = 0.5$)	626.11 ± 2.68	293.21 ± 27.22
EPOpt ($\epsilon = 0.6$)	632.16 ± 8.39	297.75 ± 39.28
EPOpt ($\epsilon = 0.7$)	*637.63 ± 11.96*	299.91 ± 35.76
EPOpt ($\epsilon = 0.8$)	636.35 ± 32.27	*308.38 ± 33.25*
EPOpt ($\epsilon = 0.9$)	628.49 ± 14.27	288.57 ± 29.23
UDR	629.43 ± 15.26	296.47 ± 28.78
RSDR ($\beta = -20$)	$\mathbf{671.70 \pm 10.07}$	414.28 ± 25.32
RSDR ($\beta = -30$)	660.59 ± 14.53	$\mathbf{432.03 \pm 42.81}$

Table 9: End-to-end wall-clock training time under matched effective PPO-update batch sizes (mean $\pm$ standard deviation over three seeds). The percentage columns show additional training time relative to UDR.

Task	UDR (s)	RSDR (s)	EPOpt (s)	RSDR vs UDR	EPOpt vs UDR
CheetahRun	1400.24 ± 2.45	1409.26 ± 2.23	1965.66 ± 2.57 ($\epsilon = 0.4$)	$+0.64\%$	$+40.38\%$
WalkerWalk	1396.15 ± 6.51	1419.19 ± 3.23	1472.91 ± 5.56 ($\epsilon = 0.8$)	$+1.65\%$	$+5.50\%$

To model agent-dependent adversarial drift, we further consider a projected Langevin-type process that biases the parameter evolution toward vulnerable regions of the learned sampler distribution:

$$\xi_{t+1} = \text{Proj}_{\Xi}(\xi_t + \alpha \nabla_{\xi} \log \nu_{\phi}(\xi_t) + \sigma \epsilon_t), \qquad \epsilon_t \sim \mathcal{N}(0, \mathbf{I}), \tag{35}$$

where α and σ independently control the drift and diffusion terms. Because ν_{ϕ} is a tanh-transformed GMM, its log-density is singular on the faces of Ξ. We therefore evaluate the inverse tanh map and its log-Jacobian after clipping the normalized coordinate to $[-1 + 10^{-6}, 1 - 10^{-6}]$, so the score is finite on all of Ξ and vanishes for coordinates on a face, which are then moved only by the noise term.

Unlike generic non-stationary perturbations based only on Gaussian or uniform noise, this process uses the *score function of the learned RSDR sampler* ν_{ϕ} to steer the dynamics toward configurations that are difficult for the trained policy. Under a risk-averse setting ($\beta < 0$), the sampler places greater mass on low-return regions, so the induced Langevin dynamics acts as a proxy for an adversarial non-stationary attack. This design is related in spirit to adversarial robustness frameworks such as TC-MDP [Zouitine et al., 2024], but avoids training a separate online adversary by instead leveraging the learned risk-sensitive distribution.

It is important to note that ν_{ϕ} is used as surrogate potential over the domain parameters. Since the sampler is shaped by the expected return, it captures regions of the parameter space where that learned policy is most likely to fail; see Figure 9 for representative trajectories.

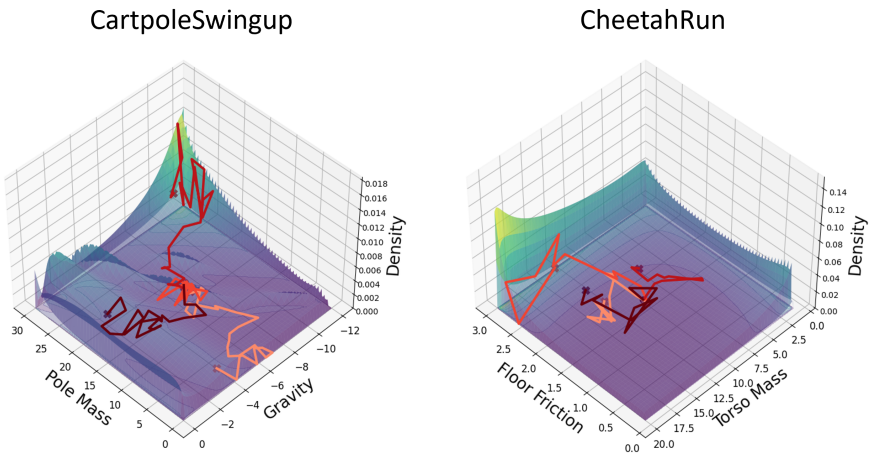


Figure 9: Visualization of Langevin Dynamics perturbations for non-stationary evaluation on CartpoleSwingup/CheetahRun task. We display four independent Langevin Dynamics for each task. The trajectories explore and gravitate toward the modes of the RSDR sampler $\nu_\phi(\xi)$, which, under a risk-averse setting ($\beta < 0$), concentrates on adversarial parameter configurations that challenge the policy.

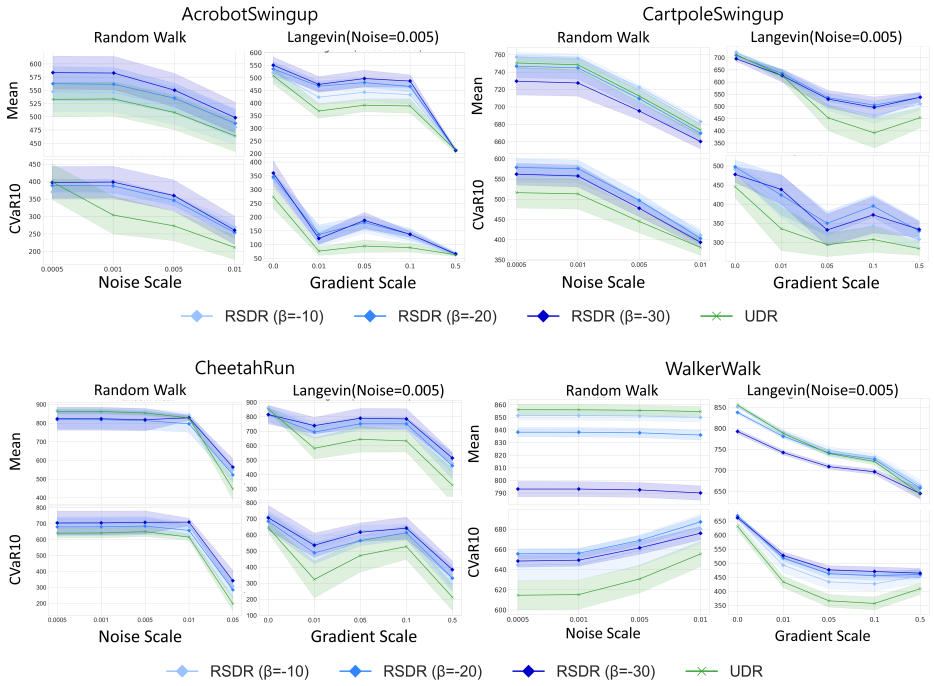


Figure 10: Non-stationary evaluation results across four benchmarks. Top rows indicate average return; bottom rows indicate CVaR_{10}. RSDR consistently demonstrates superior robustness against both Gaussian Random Walks and Adversarial Langevin Dynamics, generally outperforming the UDR baseline in both tail-risk resilience and mean performance.

For statistical reliability, we evaluate both UDR and RSDR over 10 independent training seeds. For each seed, we run 4,096 non-stationary trajectories under each attack process, and report both mean return and CVaR_{10}. As shown in Figure 10, RSDR consistently exhibits stronger robustness than UDR under both random drift and adversarially guided parameter evolution.

E.5 Benchmark results with DIS-LV

Table 10: Comparison with DIS-LV across MuJoCo Playground Benchmarks (Manipulation)

(a) PandaOpenCabinet

β	**sampler choice**	**Mean**	**CVaR$_{20}$**	**CVaR$_{10}$**	**Min**
-30	DIS-LV	163.88 ± 17.25	131.21 ± 9.77	125.05 ± 7.67	103.56 ± 6.58
	GMMVI	173.53 ± 5.83	135.84 ± 4.13	128.03 ± 3.29	105.54 ± 6.22
-20	DIS-LV	171.79 ± 5.16	130.72 ± 9.48	124.41 ± 7.46	108.12 ± 3.75
	GMMVI	177.51 ± 1.59	**142.08 ± 2.88**	**133.30 ± 2.99**	107.28 ± 5.78
-10	DIS-LV	173.07 ± 7.93	132.43 ± 10.26	125.54 ± 8.03	106.10 ± 5.01
	GMMVI	179.16 ± 4.10	140.91 ± 6.31	132.47 ± 5.90	**111.10 ± 1.95**
0	DIS-LV	167.07 ± 18.64	125.38 ± 9.88	119.85 ± 7.46	104.58 ± 3.97
	GMMVI	171.13 ± 4.90	123.56 ± 7.68	119.02 ± 5.71	105.10 ± 4.58
5	DIS-LV	171.88 ± 5.36	125.36 ± 9.10	120.06 ± 6.67	104.44 ± 4.40
	GMMVI	169.93 ± 6.06	123.27 ± 7.99	118.83 ± 5.45	105.01 ± 4.10
-	UDR	171.63 ± 3.29	122.22 ± 2.56	117.48 ± 1.29	107.53 ± 1.32

(b) PandaPickCubeOrientation

β	**sampler choice**	**Mean**	**CVaR$_{20}$**	**CVaR$_{10}$**	**Min**
-30	DIS-LV	210.69 ± 3.66	**173.15 ± 4.44**	**161.56 ± 4.22**	108.94 ± 11.92
	GMMVI	209.15 ± 9.77	166.81 ± 11.49	154.00 ± 10.42	116.60 ± 8.11
-20	DIS-LV	214.70 ± 1.33	171.34 ± 2.37	157.92 ± 2.66	104.31 ± 20.95
	GMMVI	218.38 ± 2.63	170.90 ± 5.46	156.38 ± 4.91	117.91 ± 4.77
-10	DIS-LV	221.17 ± 2.15	169.82 ± 2.01	153.55 ± 1.70	**118.47 ± 2.66**
	GMMVI	220.06 ± 3.11	163.01 ± 6.41	146.49 ± 5.74	115.34 ± 4.11
0	DIS-LV	220.04 ± 2.25	154.06 ± 3.59	138.69 ± 2.90	116.49 ± 1.46
	GMMVI	218.97 ± 2.73	152.26 ± 3.14	137.23 ± 2.85	114.59 ± 2.71
5	DIS-LV	218.35 ± 2.60	144.62 ± 4.16	131.26 ± 2.44	114.46 ± 1.34
	GMMVI	214.50 ± 3.18	137.33 ± 3.01	127.45 ± 1.05	114.10 ± 1.82
-	UDR	219.09 ± 4.85	151.56 ± 7.46	137.35 ± 4.83	116.52 ± 2.06

We compare our default sampler (GMMVI) with the diffusion-based alternative DIS-LV(Appendix D.2) in Table 10 and 11. DIS-LV also improves tail robustness over UDR in several settings, but its performance is less consistent than GMMVI across tasks. In particular, GMMVI is stronger on most DM Control tasks, while DIS-LV is competitive or better on some manipulation metrics. The choice of sampler and objective can affect the robust performance of RSDR, even though both methods aim to approximate the same intractable target distribution. Investigating principled criteria for selecting high-performing samplers in RSDR remains an important direction for future work.

Table 11: Comparison with DIS-LV across MuJoCo Playground benchmarks (DM Control Suite)

(a) AcrobotSwingup

β	sampler choice	Mean	CVaR_{20}	CVaR_{10}	Min
-30	DIS-LV	503.11 ± 69.47	417.35 ± 87.46	389.02 ± 86.90	190.24 ± 57.79
	GMMVI	**579.49 ± 36.11**	**514.23 ± 43.85**	**496.01 ± 45.81**	**383.93 ± 59.77**
-20	DIS-LV	511.94 ± 53.08	418.93 ± 73.25	386.38 ± 73.96	190.36 ± 51.18
	GMMVI	559.61 ± 31.91	498.42 ± 32.61	479.26 ± 33.39	354.05 ± 39.80
-10	DIS-LV	524.12 ± 66.94	417.43 ± 90.95	375.33 ± 97.78	164.87 ± 56.24
	GMMVI	545.34 ± 34.92	477.34 ± 34.04	455.12 ± 32.51	306.50 ± 37.12
0	DIS-LV	474.45 ± 55.25	361.08 ± 55.44	320.86 ± 61.99	156.60 ± 32.08
	GMMVI	486.95 ± 70.06	355.90 ± 93.62	312.35 ± 86.54	138.46 ± 47.14
5	DIS-LV	502.39 ± 81.90	367.98 ± 83.89	307.77 ± 93.12	111.14 ± 36.14
	GMMVI	471.87 ± 57.81	243.59 ± 99.58	199.42 ± 90.68	96.72 ± 25.22
-	UDR	532.27 ± 30.52	432.32 ± 38.84	386.42 ± 48.05	170.02 ± 47.82

(b) CartpoleSwingup

β	sampler choice	Mean	CVaR_{20}	CVaR_{10}	Min
-30	DIS-LV	730.20 ± 33.87	568.20 ± 67.96	512.40 ± 74.97	286.23 ± 72.82
	GMMVI	729.40 ± 15.92	600.76 ± 27.72	561.94 ± 28.94	**337.47 ± 38.53**
-20	DIS-LV	715.39 ± 57.28	530.12 ± 105.83	476.71 ± 106.10	300.41 ± 79.23
	GMMVI	745.43 ± 12.70	**614.34 ± 17.48**	**570.84 ± 19.43**	312.58 ± 30.24
-10	DIS-LV	725.08 ± 51.54	539.87 ± 92.45	472.59 ± 107.67	286.87 ± 48.67
	GMMVI	755.93 ± 5.90	609.24 ± 16.45	555.68 ± 24.86	310.81 ± 37.43
0	DIS-LV	716.67 ± 42.23	492.66 ± 104.71	421.01 ± 104.71	251.39 ± 56.99
	GMMVI	757.54 ± 10.71	585.03 ± 26.16	513.74 ± 40.93	286.97 ± 44.19
5	DIS-LV	713.33 ± 36.13	479.13 ± 80.54	402.30 ± 82.69	240.53 ± 51.97
	GMMVI	736.94 ± 12.92	548.53 ± 19.14	473.87 ± 23.54	276.19 ± 25.92
-	UDR	746.65 ± 7.62	570.62 ± 25.25	499.04 ± 41.93	283.61 ± 41.58

(c) CheetahRun

β	sampler choice	Mean	CVaR_{20}	CVaR_{10}	Min
-30	DIS-LV	812.15 ± 107.61	697.70 ± 117.80	659.63 ± 114.28	463.73 ± 86.46
	GMMVI	846.02 ± 22.66	**750.83 ± 30.74**	**722.14 ± 32.45**	**563.43 ± 52.16**
-20	DIS-LV	794.41 ± 71.07	655.02 ± 74.13	613.23 ± 69.58	422.54 ± 81.85
	GMMVI	848.40 ± 49.12	749.61 ± 54.24	714.02 ± 52.11	546.08 ± 64.00
-10	DIS-LV	800.11 ± 107.61	643.90 ± 139.41	589.12 ± 138.22	378.53 ± 110.13
	GMMVI	868.03 ± 20.01	747.90 ± 34.97	700.95 ± 35.29	503.69 ± 59.58
0	DIS-LV	729.16 ± 279.59	563.85 ± 236.27	501.97 ± 214.99	259.22 ± 157.04
	GMMVI	856.25 ± 24.89	683.59 ± 48.48	626.40 ± 45.37	410.39 ± 71.85
5	DIS-LV	736.06 ± 256.85	539.65 ± 194.24	463.73 ± 165.66	197.46 ± 111.54
	GMMVI	847.21 ± 13.03	630.01 ± 33.97	557.23 ± 40.92	227.48 ± 115.30
-	UDR	866.24 ± 12.54	703.71 ± 33.65	641.75 ± 33.99	446.00 ± 71.00

(d) WalkerWalk

β	sampler choice	Mean	CVaR_{20}	CVaR_{10}	Min
-30	DIS-LV	838.36 ± 6.68	719.83 ± 7.82	668.99 ± 8.19	397.71 ± 32.22
	GMMVI	794.82 ± 12.46	695.65 ± 15.24	660.59 ± 14.53	**432.03 ± 42.81**
-20	DIS-LV	850.63 ± 3.83	723.32 ± 6.79	665.78 ± 8.85	371.08 ± 26.37
	GMMVI	837.52 ± 6.34	721.47 ± 9.68	**671.70 ± 10.07**	414.28 ± 25.32
-10	DIS-LV	856.47 ± 3.38	717.26 ± 7.92	652.32 ± 10.17	331.07 ± 30.42
	GMMVI	853.59 ± 3.52	**723.69 ± 7.86**	664.50 ± 9.36	362.36 ± 40.63
0	DIS-LV	856.05 ± 4.15	697.44 ± 10.00	618.91 ± 13.14	256.35 ± 31.09
	GMMVI	857.89 ± 5.41	707.90 ± 11.72	634.76 ± 14.33	284.45 ± 22.31
5	DIS-LV	855.86 ± 4.20	689.39 ± 13.45	607.12 ± 17.80	263.00 ± 35.45
	GMMVI	856.31 ± 5.64	678.30 ± 22.41	586.70 ± 32.29	234.76 ± 25.35
-	UDR	856.91 ± 4.66	703.56 ± 11.82	629.43 ± 15.26	296.47 ± 28.78

F MuJoCo Playground benchmark

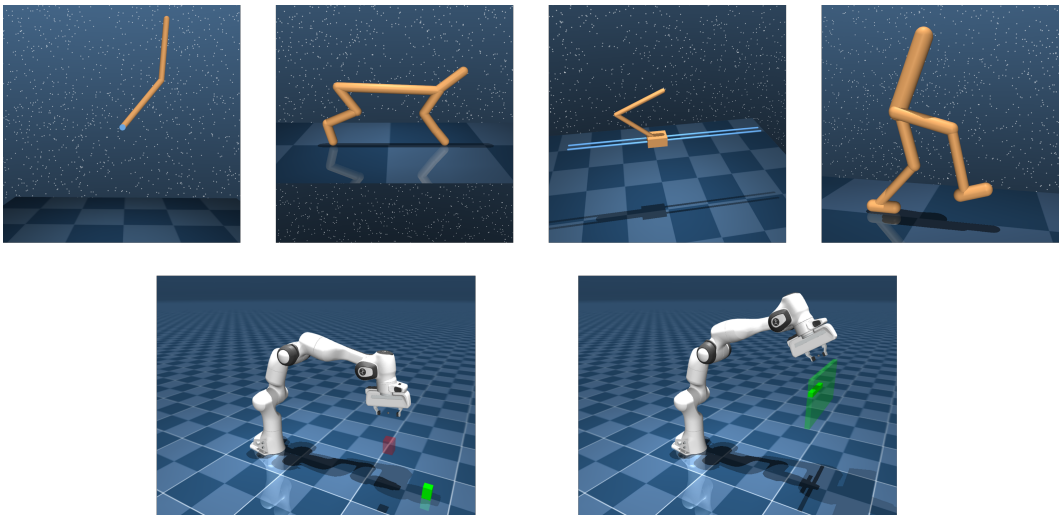


Figure 11: MuJoCo Playground Benchmark Tasks

We evaluate domain randomization algorithms on MuJoCo Playground [Zakka et al., 2025]. Like other GPU-parallelized simulators [Makoviychuk et al., 2021, Mittal et al., 2025], it allows training over many dynamics parameters in parallel at little additional wall-clock cost, which makes it well suited to domain randomization.

We employ an asymmetric critic architecture in Pinto et al. [2017a], which leverages privileged information—for our case the latent dynamics parameters ξ—to provide more accurate value estimates. This approach is standard practice for robust Sim2Real transfer in robotics [Akkaya et al., 2019, Eschmann et al., 2024, Lin et al., 2025, Zakka et al., 2025, Dong et al., 2026] and ensures consistency with all of our baseline curriculum DR algorithms [Akkaya et al., 2019, Tiboni et al., 2024, Curtis et al., 2025].

For both RSDR and GoFlow, we normalize each episodic return by the maximum possible episode length. Together with the normalized per-step rewards, this places returns on a comparable numerical scale across tasks. This facilitates the use of common temperature search grids, although the preferred temperature can still depend on the task and evaluation metric. Baseline hyperparameter selection is described in Appendix G.2.

The main benchmark return metrics are computed over 10 independent training seeds. For evaluation, we test the resulting policies across a set of dynamics parameters Ξ. For $\xi \in \mathbb{R}^2$(AcrobotSwingup, CartpoleSwingup, CheetahRun), we evaluated on a 64×64 grid yielding 4096 unique dynamics parameter configurations.

For high dimensional tasks(WalkerWalk, PandaPickCubeOrientation, PandaOpenCabinet), 16384 dynamics parameters sampled from a uniform distribution are used. For each parameter configuration, the return is calculated as the average episode reward across 10 evaluation seeds to account for stochasticity in the environment. All experiments were conducted on a 3.00GHz Intel Core i9-10980XE CPU, 64G RAM, and NVIDIA GeForce RTX 3090 GPU.

The specific dynamics parameter range is represented in Table 12.

Table 12: Domain randomization range Ξ for each task

Dynamics Parameter	Range Ξ
AcrobotSwingup(2D)	
Upper Arm Mass	$[0.78, 3.0]$
Lower Arm Mass	$[0.5, 1.2]$
CartpoleSwingup(2D)	
Pole Mass	$[0.01, 30.0]$
Gravity	$[-12.0, 0.0]$
CheetahRun(2D)	
Floor Friction	$[0.1, 3.0]$
Torso Mass	$[0.8, 20.0]$
WalkerWalk(15D)	
Floor Friction	$[0.82, 7.58]$
Torso Mass	$[0.88, 6.19]$
Thigh Mass(Left, Right)	$[0.98, 6.87]$
Leg Mass(Left, Right)	$[0.68, 4.75]$
Foot Mass(Left, Right)	$[0.74, 5.15]$
Body offset(7D)	$[-0.05, 0.05]$
PandaPickCubeOrientation(29D)	
Gripper Friction	$[0.3, 3.0]$
Cube Mass	$[0.1, 8.0]$
Franka Body Mass(11D)	$[0.8, 1.2]$
Joint Damping(9D)	$[0.8, 1.2]$
Actuator Gain(7D)	$[0.9, 1.1]$
PandaOpenCabinet(30D)	
Gripper Friction	$[0.3, 10.0]$
Cabinet Friction	$[0.3, 10.0]$
Handle Mass	$[0.1, 20.0]$
Franka Body Mass(11D)	$[0.8, 1.2]$
Joint Damping(9D)	$[0.8, 1.2]$
Actuator Gain(7D)	$[0.7, 1.3]$

G Implementation details

G.1 Target distribution equation of the 2D domain example in Figure 2

We set the target distribution in Figure 2 to be $p_\beta = e^{\beta E}/Z_\beta$, where

$$E([x;y]) = \\ -\tanh\left(\exp\left(-\frac{1}{2}\left(\frac{\sqrt{(2x-1)^2+(2y-1)^2}-d_0}{s_r}\right)^2\right)\cos\left(m\,\mathrm{atan2}\left(2y-1,2x-1\right)\right)\right), \tag{36}$$

where m is the number of maxima (or minima), d_0 is the diameter of the circle centered at $(0.5, 0.5)$ on which the maxima (or minima) lie, and s_r is a variance coefficient.

G.2 Configuration and hyperparameter choices of baseline domain randomization methods

The success threshold J_T for AutoDR [Akkaya et al., 2019], and DORAEMON [Tiboni et al., 2024] is set to 0.7 (with respect to the normalized rewards) for AcrobotSwingup, CartpoleSwingup, CheetahRun, and WalkerWalk, and 0.5 for PandaPickCubeOrientation and PandaOpenCabinet.

We tune DORAEMON and GoFlow [Curtis et al., 2025] hyperparameters for each task. For DORAEMON, we first search over the trust region parameter $\epsilon_D \in [0.005, 0.01, 0.5]$ and search for the success rate condition $\alpha_D \in [0.5, 0.7, 0.8]$. For GoFlow, we first search over the entropy coefficient $\alpha_G \in [0.5, 1.0, 1.5, 2.0]$ and subsequently optimize the KL-divergence regularization term $\gamma \in \{0.0, 0.5, 1.0, 2.0\}$ for each α_G configuration.

Note that these hyperparameters were tuned to maximize training stability and curriculum learning efficiency rather than directly optimizing for tail-end metrics such as CVaR_{10} or minimum return. Consequently, we selected the optimal configurations based on the highest aggregate mean performance across the entire dynamics parameter space Ξ.

We implemented GoFlow with the JAX-based bijx [Gerdes and Cheng, 2025] library. Specifically, we reimplemented neural spline flows [Durkan et al., 2019] and replaced the affine transformations of standard masked autoregressive flows (MAF) [Papamakarios et al., 2017] with rational-quadratic splines [Durkan et al., 2019].

In our experiments, we use Brax [Freeman et al., 2021] PPO implementation as it shows higher performance on MuJoCo Playground benchmark than other RL libraries. The specific hyperparameter setting follows their GitHub repository `https://github.com/google-deepmind/mujoco_playground/`.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state the paper's main claims: a KL-DRO formulation of episodic domain randomization over dynamics-parameter distributions, the resulting exponentially tilted policy-dependent sampler, and an amortized sampler-learning algorithm. Section 4 presents the formulation and theoretical characterization, Section 5 presents the empirical results, and Section 6 states the main limitations.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 6 discusses the main limitations: the focus on the risk-averse regime ($\beta < 0$), the fixed-β amortized alternating approximation rather than an exact fixed-radius minimax solver, and the simulation-only nature of the evaluation.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The main theoretical results are presented in Section 4. Appendix A states the regularity assumptions, proves the KL-DRO soft-adversary result, and analyzes the relationship between the KL radius and the temperature parameter β; Appendix B gives the entropic-risk duality and the symmetric optimistic counterpart for $\beta > 0$.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide open-source code and describe the experimental setup, task randomization ranges, evaluation protocol, sampler implementation, and hyperparameter choices in Appendices D–G. Algorithm 1 summarizes the training procedure used to reproduce the proposed method.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: An open-source code repository is provided at `https://github.com/jeongsukchul/risk-sensitive-domain-randomization`. The repository includes installation instructions and scripts/configurations for reproducing the main experiments.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 describes the benchmark tasks, evaluation metrics, baselines, and seed protocol. Appendices D–G provide sampler details, experimental protocols, task randomization ranges, implementation details, and baseline hyperparameter choices.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Section 5 and Appendix F specify the number of independent training seeds and evaluated dynamics parameters. Tables and shaded regions report mean $\pm$ one standard deviation over 10 independent training seeds.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix F reports the hardware used for the experiments, including CPU, RAM, and GPU specifications.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and believe that the research conducted in this paper conforms to it.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The work may have positive impact by improving robustness of policies trained under simulator-parameter uncertainty. A potential negative impact is over-reliance on simulation-only robustness as evidence for safe real-world deployment; we therefore emphasize in Section 6 that physical validation remains necessary before deployment.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: The paper does not release high-risk models, datasets, or assets such as pretrained generative models, scraped datasets, or systems requiring special safeguards for controlled access.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We cite the papers and repositories corresponding to the existing simulators, algorithms, and codebases used in our experiments. The released repository documents the relevant licenses and usage terms for these assets, and our use follows those terms.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The new asset introduced by this paper is the open-source code release. The repository includes a README with installation instructions, experiment scripts/configurations, and documentation for reproducing the main results.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: This work does not involve crowdsourcing, user studies, or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This work does not involve crowdsourcing, user studies, or research with human subjects, so IRB approval or an equivalent human-subjects review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development, experiments, and scientific claims do not use LLMs as an important, original, or non-standard component. Any LLM use was limited to language editing or formatting assistance and did not affect the methodology, scientific rigor, or originality of the research.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.