GENIEO: Tractable Maximum-Entropy Off-Policy Learning with Invertible Generative Policies

Feihong Zhang[1,†], Guojian Zhan[1,†], Yinuo Wang[2], Letian Tao[1], Likun Wang[3]
Shiqi Liu[1], Yao Lyu[1,*], Shengbo Eben Li[1,*]

[1] College of AI & School of Vehicle and Mobility, Tsinghua University
[2] 2030Lab, Yinwang Intelligent Technology
[3] National University of Singapore

[†] Equal contribution. [*] Corresponding authors.

Abstract

Generative models, such as *diffusion* and *flow-based models*, have emerged as powerful policy approximators in deep reinforcement learning (RL) due to their ability to represent multi-modal action distributions. However, their integration into maximum entropy RL remains challenging, as some existing methods treat generative processes as black-box samplers or rely on heuristic noise. In this paper, we propose GENIEO, a novel off-policy generative RL framework that leverages the tractable density of a paired augmented-action policy for entropy-regularized learning. By designing a structurally invertible affine dual-variable flow, we avoid the need for numerical ODE inversion. This design allows us to rigorously apply the change-of-variables formula to derive exact augmented-action log-densities and Monte Carlo entropy estimates, enabling their direct integration into the off-policy learning objective. Experimental evaluations across 14 challenging tasks from the DMControl and HumanoidBench suites demonstrate that GENIEO achieves the highest observed suite-average returns among the evaluated generative-policy baselines. By supporting entropy-regularized policy learning in complex, high-dimensional environments, our approach bridges the gap between expressive generative modeling and maximum entropy RL.

1 Introduction

Reinforcement learning continues to advance through improved policy optimization [1], stability-oriented optimization [2], entropy-aware learning [3], efficient generative policies [4], and model-based policy improvement and planning [5, 6]. Applications to safe and efficient autonomous driving [7] and real-time driving decisions [8] further motivate expressive policy representations for complex control tasks. Generative models [9, 10, 11, 12], such as diffusion [13] and flow-based models [14], provide an expressive policy class for such tasks. In contrast to conventional Gaussian policies, generative models are capable of representing complex and multi-modal action distributions, making them suitable for challenging robotic manipulation and high-dimensional control tasks. Recent studies, including FlowRL [15], DIPO [16], DACER [17], QVPO [18], and DIME [19], have demonstrated the potential of training generative policies in off-policy settings. However, while these models offer superior representational capacity, ensuring effective exploration remains a fundamental challenge.

The maximum entropy framework maintains policy stochasticity by regularizing the learning objective with policy entropy. Under traditional Gaussian policies, this principle has catalyzed foundational

*Corresponding authors: lyo.tobias@foxmail.com; lishbo@tsinghua.edu.cn.

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

algorithms ranging from Soft Actor-Critic (SAC) [20] to advanced refinements such as DSAC-T [21] and DSAC-E [3]. However, maximizing the entropy of simple parametric distributions frequently struggles in sparse-reward or highly non-convex action landscapes, where the policy easily collapses to local optima. While generative models naturally provide richer behavioral diversity to navigate such complex settings, existing off-policy generative methods [15, 16, 17, 18] typically treat the generative process as a black-box sampler or rely on heuristic noise injection to introduce external randomness. Consequently, these approaches largely overlook the intrinsic probabilistic structure of generative policies, leaving their principled integration into the maximum entropy framework not yet fully realized.

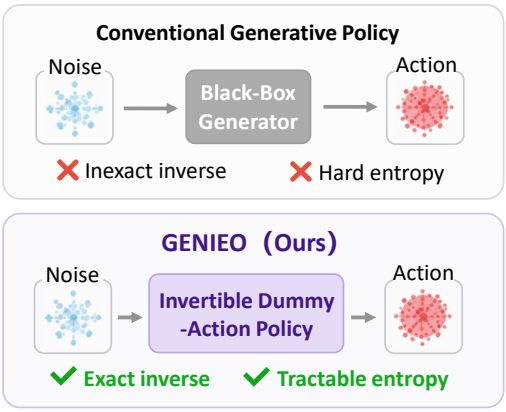


Figure 1: **Overview of GENIEO.** An invertible augmented-action policy provides tractable sample log-densities for Monte Carlo entropy estimation and off-policy MaxEnt learning.

In this paper, we propose GENIEO, a novel off-policy generative RL framework that uses tractable entropy regularization for off-policy learning, as illustrated in Figure 1. Our key insight is that the intrinsic entropy of a generative policy, when estimated by Monte Carlo in the augmented dummy-action space, can serve as a principled policy regularizer. To avoid numerical ODE inversion, we introduce a structurally invertible affine dual-variable architecture. This design enables exact reversibility and allows us to apply the change-of-variables formula to derive the augmented dummy-action policy likelihood, entropy, and entropy gradient. As a result, intrinsic entropy regularization can be directly integrated into the off-policy learning objective with replay buffer data.

We evaluate GENIEO across 14 challenging tasks from the DMControl and HumanoidBench suites. Experimental results demonstrate that our framework achieves the highest observed performance among the evaluated generative-policy baselines in complex environments where conventional Gaussian policies and heuristic generative approaches struggle. By learning high-return policies in these high-dimensional settings, our approach bridges the gap between expressive generative modeling and maximum entropy RL.

The main contributions of this work are three-fold:

- We introduce GENIEO, a novel off-policy generative RL framework that leverages tractable entropy regularization of an augmented dummy-action policy for off-policy learning.
- We provide a rigorous mathematical derivation for computing the exact dummy-action policy likelihood and Monte Carlo estimates of entropy and its gradient within a structurally invertible affine flow-based architecture, bridging generative modeling and maximum entropy RL.
- We demonstrate through extensive experiments that our method achieves the highest observed mean returns among the evaluated generative-policy baselines across 14 challenging continuous control tasks from the DMControl and HumanoidBench suites.

2 Preliminary

Maximum Entropy Reinforcement Learning. We consider a Markov Decision Process (MDP) defined by $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma \rangle$, where $\gamma \in [0, 1)$ is the discount factor [22]. In contrast to standard RL, maximum entropy RL [23] encourages exploration by augmenting the reward with a policy entropy term. The objective is to maximize the expected discounted return and entropy:

$$J(\pi) = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t) + \alpha \mathcal{H}(\pi(\cdot|s_t)) \right) \right], \tag{1}$$

where α is the temperature coefficient and $\mathcal{H}(\pi(\cdot|s_t)) = \mathbb{E}_{a_t \sim \pi}[-\log \pi(a_t|s_t)]$ denotes the policy entropy. The optimal policy is typically found via *Soft Policy Iteration*, which alternates between two stages:

Soft Policy Evaluation (PEV) aims to estimate the soft Q-value, $Q^{\text{soft}}(s,a)$, for a fixed policy π. This is achieved by repeatedly applying the soft Bellman operator $\mathcal{T}^{\pi}$:

$$\mathcal{T}^{\pi}Q^{\text{soft}}(s,a) = r(s,a) + \gamma \mathbb{E}_{s' \sim \mathcal{P}}[V^{\text{soft}}(s')], \tag{2}$$

where the soft state value function $V^{\text{soft}}(s')$ is defined as:

$$V^{\text{soft}}(s) = \mathbb{E}_{a \sim \pi}[Q^{\text{soft}}(s,a) - \alpha \log \pi(a|s)]. \tag{3}$$

Soft Policy Improvement (PIM) updates the policy toward the exponential of the current soft Q-function. The new policy π_{new} is obtained by minimizing the Kullback-Leibler (KL) divergence between the policy and the Boltzmann distribution induced by Q^{soft}:

$$\pi_{\text{new}} = \arg\min_{\pi' \in \Pi} \mathrm{D}_{\mathrm{KL}}\left(\pi'(\cdot|s) \middle\| \frac{\exp(Q^{\text{soft}}(s,\cdot)/\alpha)}{Z^{\text{soft}}(s)}\right), \tag{4}$$

where $Z^{\text{soft}}(s)$ is the normalizing partition function. This improvement step ensures that $V^{\pi_{\text{new}}}(s) \geq V^{\pi_{\text{old}}}(s)$ for all $s \in \mathcal{S}$.

Generative models as RL policies. In reinforcement learning, a generative policy $\pi_\theta(a|s)$ represents the action distribution by transforming a latent noise $z_0 \sim p(z_0)$ (e.g., $\mathcal{N}(\mathbf{0}, \mathbf{I})$) into an executable action $a = z_1$ conditioned on the state s. Under the flow matching framework [14], this transformation is defined by a continuous-time velocity field $v_\theta(z_t, t, s)$, and the generation process follows the continuous-time flow ODE:

$$\frac{dz_t}{dt} = v_\theta(z_t, t, s), \quad t \in [0,1]. \tag{5}$$

To align the generative distribution with target actions or policy-improvement signals, the velocity field is optimized using the conditional flow matching objective:

$$\mathcal{L}(\theta) = \mathbb{E}_{t,z_0,a}\left[\|v_\theta(z_t,t,s) - (a - z_0)\|^2\right], \tag{6}$$

where $z_t = (1-t)z_0 + ta$ and $t \sim \mathcal{U}[0,1]$. However, discrete numerical solvers generally do not provide a closed-form inverse by simply reversing the update, making exact latent recovery and tractable entropy computation non-trivial, as discussed in Section 3.1.

3 Method

This section describes GENIEO, a dual-variable affine generative policy that enables exact inversion and tractable dummy-action likelihood evaluation for maximum entropy off-policy RL. We detail the invertible architecture, establish its entropy computation, and outline the RL implementation.

3.1 Generative Policy Structure with Tractable Entropy

Continuous-time flow policies implemented with numerical solvers often face difficulties in exact entropy calculation due to the numerical asymmetry between generation and inversion. As shown in Eq. (7), although the underlying flow defines a continuous-time evolution, practical generation and inversion are typically approximated by discrete solvers such as Euler integration. During generation, the solver advances the state using $v_\theta(\mathbf{z}_t, t, s)$. However, the corresponding reverse Euler step evaluated at $\mathbf{z}_{t+\Delta t}$ does not generally recover the original $\mathbf{z}_t$. This discretization gap prevents closed-form exact inversion by simply reversing the numerical solver, making exact latent recovery and likelihood evaluation non-trivial.

$$\begin{aligned} \text{Generative (Euler Step):} \quad & \mathbf{z}_{t+\Delta t} = \mathbf{z}_t + \Delta t\, v_\theta(\mathbf{z}_t, t, s), \\ \text{Inversion (Naive Reverse Step):} \quad & \mathbf{z}_t \approx \mathbf{z}_{t+\Delta t} - \Delta t\, v_\theta(\mathbf{z}_{t+\Delta t}, t+\Delta t, s). \end{aligned} \tag{7}$$

To address this, we propose GENIEO, which employs an exactly invertible affine generative structure as illustrated in Figure 2. The discrete-time affine alternating updates use established normalizing-flow coupling tools [24, 25] to obtain bijectivity and tractable log-determinants. The framework consists of the following key components:

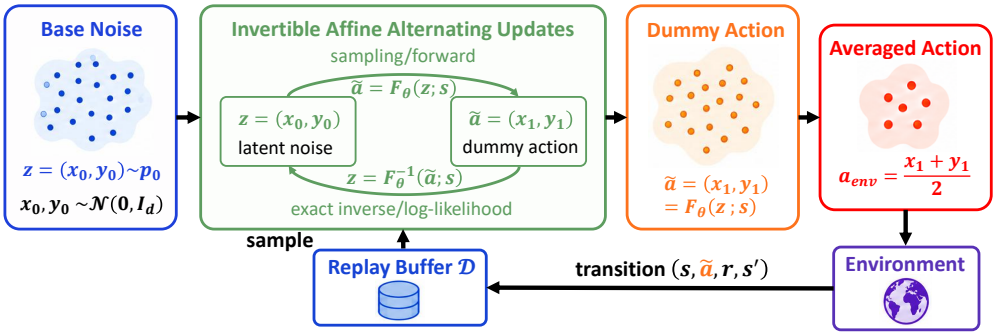


Figure 2: **GENIEO Policy Architecture.** The architecture illustrates the process of generating augmented dummy actions through a structurally invertible flow, followed by averaging for environment interaction. The process includes both forward sampling and exact inverse steps for likelihood computation.

Space Expansion. To enable an exactly invertible coupling structure, GENIEO defines the policy distribution in a doubled dummy-action space. Instead of directly generating an environment action $a \in \mathbb{R}^d$, the policy first generates an augmented dummy action $\tilde{a} = (x, y) \in \mathbb{R}^{2d}$, where $x, y \in \mathbb{R}^d$.

Latent Initialization. Given a state s, two independent Gaussian noises $x_0, y_0 \in \mathbb{R}^d$ are sampled from $\mathcal{N}(0, I_d)$. Starting from (x_0, y_0), the policy applies a sequence of invertible affine alternating updates from flow time $t = 0$ to $t = 1$, yielding the terminal dummy action $(x_1, y_1) = F_\theta(x_0, y_0; s)$.

Alternating Update. For each discretization step $t \to t + \Delta t$, the variables are updated through a composition of two triangular affine coupling transformations:

$$\begin{aligned} X &= x_t \odot \exp\big(\Delta t\, \ell_\theta(y_t, t, s)\big) + \Delta t\, u_\theta(y_t, t, s), \\ Y &= y_t \odot \exp\big(\Delta t\, \rho_\theta(X, t, s)\big) + \Delta t\, w_\theta(X, t, s). \end{aligned} \tag{8}$$

The updated variables are then set as $(x_{t+\Delta t}, y_{t+\Delta t}) = (X, Y)$. This affine coupling ensures the first equation updates x using the previous value of y, while the second updates y using the newly obtained X, preserving exact invertibility and yielding a tractable log-determinant.

Interaction with the Environment. Although the policy density is defined over the terminal dummy action $\tilde{a} = (x_1, y_1)$, the environment receives only the projected action $a = (x_1 + y_1)/2$. This many-to-one projection does not preserve entropy: with $b = (x_1 - y_1)/2$, $\mathcal{H}(x_1, y_1 \mid s) = \mathcal{H}(a \mid s) + \mathcal{H}(b \mid a, s) + d \log 2$ whenever these differential entropies are finite. The coherence penalty discourages projection-null variation in b but does not equate augmented and executed-action entropy or guarantee broader state visitation.

Policy Entropy Computation. Since the policy defines a bijective map $(x_0, y_0) \mapsto (x_1, y_1)$, the dummy-action density is computed via the change-of-variables formula:

$$\log \pi_\theta(\tilde{a} \mid s) = \log p_0(x_0, y_0) - \log \left| \det \frac{\partial F_\theta(x_0, y_0; s)}{\partial (x_0, y_0)} \right|, \tag{9}$$

where p_0 is the standard Gaussian density of the initial noise and $(x_0, y_0) = F_\theta^{-1}(\tilde{a}; s)$. For the affine alternating updates, the log-determinant is accumulated analytically from the per-step terms in Eq. 13. Accordingly, we formulate the exact intrinsic entropy of the augmented dummy-action policy as $\mathcal{H}(\pi_\theta(\cdot|s)) = -\mathbb{E}_{\tilde{a} \sim \pi_\theta(\cdot|s)}[\log \pi_\theta(\tilde{a} \mid s)]$, enabling principled likelihood evaluation without numerical ODE inversion.

3.2 Theoretical Analysis of Exact Invertibility and Generative Policy Entropy

To establish a principled foundation for exact dummy-action likelihood evaluation and Monte Carlo entropy estimation, we first recall the change-of-variables formula, which relates the density of a transformed random variable to its base distribution.

Lemma 1 (Change-of-variables [26]). *Let $f : \mathbb{R}^n \to \mathbb{R}^n$ be a C^1 diffeomorphism. If $X \sim q(x)$ and $Y = f(X)$ is the transformed random variable, then the density $p(y)$ of Y is*

$$p(y) = q(x) \left| \det \frac{\partial f(x)}{\partial x} \right|^{-1}, \qquad x = f^{-1}(y). \tag{10}$$

Proof. See Appendix A.1. □

The utility of Lemma 1 in generative modeling depends strictly on the invertibility of the transformation. We now prove that our proposed affine alternating update rule satisfies this requirement exactly, ensuring that the mapping remains bijective in exact arithmetic.

Theorem 2 (Exact Invertibility of GENIEO Affine Alternating Update). *Let s and t be fixed, $h \in \mathbb{R}$, and let $u_\theta, \ell_\theta, w_\theta, \rho_\theta : \mathbb{R}^d \to \mathbb{R}^d$ be continuously differentiable functions conditioned on (t, s), where u_θ, w_θ are shift functions and ℓ_θ, ρ_θ are log-scale functions. For $x, y \in \mathbb{R}^d$, define $F_h(x, y) = (X, Y)$ by*

$$X = x \odot e^{h\ell_\theta(y,t,s)} + h\, u_\theta(y, t, s), \qquad Y = y \odot e^{h\rho_\theta(X,t,s)} + h\, w_\theta(X, t, s), \tag{11}$$

where $\odot$ and $e^{(\cdot)}$ are applied element-wise. Then $F_h : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ is exactly invertible, with inverse

$$y = \big(Y - h\, w_\theta(X, t, s)\big) \odot e^{-h\rho_\theta(X,t,s)}, \qquad x = \big(X - h\, u_\theta(y, t, s)\big) \odot e^{-h\ell_\theta(y,t,s)}. \tag{12}$$

Moreover,

$$\log \left| \det \frac{\partial F_h(x, y)}{\partial (x, y)} \right| = h\, \mathbf{1}^\top \ell_\theta(y, t, s) + h\, \mathbf{1}^\top \rho_\theta(X, t, s). \tag{13}$$

Proof. See Appendix A.2. □

By combining the change-of-variables formula in Lemma 1 with the exact invertibility established in Theorem 2, and noting that a finite composition of diffeomorphisms remains a diffeomorphism, we can derive a tractable expression for the augmented dummy-action policy entropy. Theorem 3 further characterizes this entropy expression and its gradient with respect to the policy parameters.

Theorem 3 (Entropy and Gradient of Invertible Dummy-Action Policy). *Let GENIEO define an augmented dummy-action policy by $\tilde{a} = F_\theta(z; s)$, where $z = (x_0, y_0) \sim p_0(z)$ and $\tilde{a} = (x_1, y_1) \in \mathbb{R}^{2d}$. Assume that p_0 is independent of θ, has finite entropy $\mathcal{H}(p_0)$, and that $F_\theta(\cdot; s) : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ is a C^1 diffeomorphism obtained by finitely composing GENIEO affine alternating updates. Then the conditional entropy $\mathcal{H}_\theta(s) = -\mathbb{E}_{\tilde{a}\sim\pi_\theta(\cdot|s)}[\log \pi_\theta(\tilde{a} \mid s)]$ satisfies $\mathcal{H}_\theta(s) = \mathcal{H}(p_0) + \mathbb{E}_{z\sim p_0}\left[\log \left| \det \frac{\partial F_\theta(z;s)}{\partial z} \right|\right]$, where the log-determinant is the sum of the per-step log-determinants in Eq. 13. Moreover, under regularity conditions that allow exchanging differentiation and expectation,*

$$\nabla_\theta \mathcal{H}_\theta(s) = \mathbb{E}_{z\sim p_0}\left[\nabla_\theta \log \left| \det \frac{\partial F_\theta(z; s)}{\partial z} \right| \right]. \tag{14}$$

Proof. See Appendix A.3. □

3.3 Generative Maximum Entropy RL and Implementation

With exact dummy-action likelihood and entropy computation, GENIEO can be naturally incorporated into a maximum entropy off-policy RL framework. This section describes the resulting training procedure, including soft policy evaluation, soft policy improvement, and automatic entropy tuning.

Policy Evaluation (PEV). We learn two soft critics Q_{ϕ_1} and Q_{ϕ_2} over the augmented dummy-action space, where $Q_\phi(s, \tilde{a})$ evaluates the soft value of a dummy action whose projection is executed in the environment. For each replay transition $(s, \tilde{a}, r, s') \sim \mathcal{D}$, with $\tilde{a} = (x_1, y_1)$, the soft Bellman target is

$$\mathcal{Y} = r + \gamma \mathbb{E}_{\tilde{a}'\sim\pi_\theta(\cdot|s')}\left[\min_{i=1,2} Q_{\bar{\phi}_i}(s', \tilde{a}') - \alpha \log \pi_\theta(\tilde{a}'|s') \right]. \tag{15}$$

The critics are updated by minimizing

$$\mathcal{L}_Q(\phi_i) = \mathbb{E}_{(s,\tilde{a},r,s')\sim\mathcal{D}}\left[\big(Q_{\phi_i}(s, \tilde{a}) - \mathcal{Y}\big)^2 \right], \qquad i = 1, 2. \tag{16}$$

Policy Improvement (PIM). Given the learned critics, we update the generative policy by reparameterized sampling from the base distribution. Specifically, for $z \sim p_0(z)$, the actor deterministically generates the terminal dummy action $\tilde{a}_\theta(s,z) = (x_1, y_1) = F_\theta(z; s)$, allowing gradients from the critic and log-determinant entropy terms to be backpropagated through the affine flow. The actor is optimized by minimizing

$$\mathcal{L}_\pi(\theta) = \mathbb{E}_{s\sim\mathcal{D},\, z\sim p_0}\left[-\min_{i=1,2} Q_{\phi_i}\big(s, \tilde{a}_\theta(s,z)\big) + \alpha \log \pi_\theta\big(\tilde{a}_\theta(s,z) \mid s\big) + \nu\|x_1 - y_1\|_2^2\right]. \quad (17)$$

The first term maximizes the soft value of generated dummy actions, the second applies entropy regularization through the exact dummy-action log-likelihood, and the final coherence penalty keeps the paired variables close under the averaged environment-action projection.

Automatic Entropy Tuning. Rather than treating the temperature α as a fixed hyperparameter, we dynamically adjust it to track a target entropy $\bar{\mathcal{H}}$ defined in the augmented dummy-action space, typically set to $\bar{\mathcal{H}} = -\dim(\tilde{\mathcal{A}}) = -2\dim(\mathcal{A})$. The temperature is optimized by minimizing

$$\mathcal{L}(\alpha) = \mathbb{E}_{s\sim\mathcal{D},\, \tilde{a}\sim\pi_\theta(\cdot|s)}\left[-\alpha\left(\log \pi_\theta(\tilde{a}|s) + \bar{\mathcal{H}}\right)\right]. \quad (18)$$

The complete training procedure of GENIEO is summarized in Algorithm 1.

Algorithm 1 GENIEO: Off-Policy Maximum Entropy RL with an Augmented Flow Policy

Initialize actor π_θ, critics Q_{ϕ_1}, Q_{ϕ_2}, target critics $Q_{\bar{\phi}_1}, Q_{\bar{\phi}_2}$, replay buffer $\mathcal{D}$, temperature α, coherence weight ν, and target entropy $\bar{\mathcal{H}}$
for each environment step **do**
  Observe state s
  Sample base noise $z \sim p_0(z)$ and generate terminal dummy action $\tilde{a} = (x_1, y_1) = F_\theta(z; s)$ via affine alternating updates
  Project $\tilde{a}$ to the environment action a_{env} and execute it
  Observe reward r and next state s'
  Store transition $(s, \tilde{a}, r, s')$ in $\mathcal{D}$
end for
for each gradient step **do**
  Sample a mini-batch $(s, \tilde{a}, r, s')$ from $\mathcal{D}$
  Sample next dummy action $\tilde{a}' \sim \pi_\theta(\cdot \mid s')$ and compute $\log \pi_\theta(\tilde{a}' \mid s')$
  Compute the soft Bellman target $\mathcal{Y}$ using Eq. 15
  Update critics Q_{ϕ_1}, Q_{ϕ_2} by minimizing Eq. 16
  Sample $z \sim p_0(z)$ and generate $\tilde{a}_\theta(s,z) = F_\theta(z; s)$ via affine alternating updates
  Compute $\log \pi_\theta(\tilde{a}_\theta(s,z) \mid s)$ using the exact dummy-action likelihood and affine log-determinant
  Update actor θ by minimizing Eq. 17
  Update temperature α by minimizing Eq. 18
  Update target critics by Polyak averaging: $\bar{\phi}_{1,2} \leftarrow \tau\phi_{1,2} + (1-\tau)\bar{\phi}_{1,2}$
end for

4 Experiment

4.1 Main Experiment

Benchmark. Our experimental evaluation spans 14 demanding continuous control environments to evaluate the proposed algorithm. Ten of these are selected from the DeepMind Control Suite (DMC) [27], featuring diverse behavioral modes such as *Stand*, *Walk*, *Trot*, and *Run*. These scenarios require dynamic balancing and multi-limb coordination across several intricate morphologies, including the dog, humanoid, quadruped, and walker. To push the limits of whole-body control, we also incorporate 4 challenging tasks from the recently developed HumanoidBench (H-Bench) [28] based on the Unitree H1 robot. These encompass precise spatial manipulation (*H1-reach*), complex chair navigation (*H1-sit hard*), tunnel crawling (*H1-crawl*), and postural maintenance (*H1-balance simple*). A comprehensive breakdown of these environments is available in Appendix B.1.

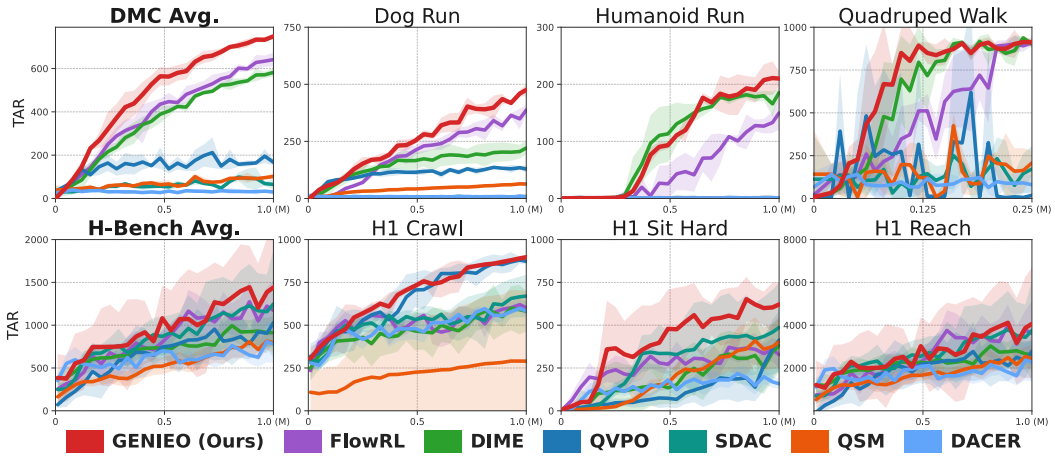


Figure 3: **Representative performance curves on DMControl and HumanoidBench.** Solid lines show mean episode returns over 5 seeds, shaded bands denote ± 1 standard deviation, and the remaining task curves are provided in Appendix C.

Table 1: Total Average Return (TAR) on 10 DMC Suite tasks and 4 H-Bench tasks. Mean $\pm$ Std over 5 seeds. **Bold** = best, underlined = second-best. Higher is better.

Task	DACER	SDAC	QSM	QVPO	DIME	FlowRL	GENIEO (Ours)
Dog Stand	22 ± 11	289 ± 174	405 ± 11	616 ± 27	559 ± 105	916 ± 105	$\mathbf{973 \pm 10}$
Dog Walk	11 ± 2	8 ± 3	108 ± 17	189 ± 30	374 ± 26	905 ± 16	$\mathbf{911 \pm 5}$
Dog Run	7 ± 5	8 ± 3	63 ± 7	129 ± 16	221 ± 31	387 ± 21	$\mathbf{476 \pm 28}$
Dog Trot	8 ± 6	12 ± 3	10 ± 2	144 ± 9	510 ± 126	745 ± 13	$\mathbf{773 \pm 25}$
Humanoid Stand	6 ± 0	6 ± 0	6 ± 0	5 ± 4	805 ± 40	431 ± 26	$\mathbf{855 \pm 10}$
Humanoid Walk	2 ± 0	1 ± 0	2 ± 0	2 ± 0	695 ± 47	421 ± 181	$\mathbf{730 \pm 6}$
Humanoid Run	1 ± 0	1 ± 0	1 ± 0	1 ± 0	186 ± 7	150 ± 29	$\mathbf{210 \pm 13}$
Quadruped Walk	80 ± 56	172 ± 135	207 ± 70	18 ± 18	903 ± 25	908 ± 18	$\mathbf{913 \pm 31}$
Walker Stand	140 ± 99	130 ± 44	192 ± 76	431 ± 139	982 ± 4	981 ± 4	$\mathbf{984 \pm 2}$
Walker Run	22 ± 6	20 ± 4	24 ± 1	138 ± 87	627 ± 24	570 ± 25	$\mathbf{661 \pm 38}$
AVG. DMC Suite	30 ± 19	65 ± 37	102 ± 18	167 ± 33	586 ± 44	641 ± 44	$\mathbf{749 \pm 17}$
H1 Reach	2274 ± 903	3757 ± 2327	2439 ± 640	2788 ± 579	2610 ± 642	3615 ± 1510	$\mathbf{4063 \pm 2589}$
H1 Sit Hard	158 ± 29	487 ± 261	403 ± 153	413 ± 171	380 ± 158	328 ± 79	$\mathbf{621 \pm 130}$
H1 Crawl	583 ± 28	670 ± 125	290 ± 350	873 ± 17	582 ± 119	602 ± 34	$\mathbf{897 \pm 14}$
H1 Balance Simple	99 ± 16	79 ± 2	53 ± 2	42 ± 5	75 ± 22	146 ± 22	$\mathbf{184 \pm 30}$
AVG. H-Bench	779 ± 244	1248 ± 679	796 ± 286	1029 ± 193	912 ± 235	1173 ± 411	$\mathbf{1441 \pm 691}$

Baselines. We compare GENIEO with six representative generative online RL algorithms based on diffusion or continuous flow policies: (1) **DACER** [17], which uses diffusion policies and GMM-estimated entropy for exploration; (2) **QSM** [29], which aligns policy scores with Q-function action gradients; (3) **QVPO** [18], which optimizes diffusion policies via a Q-weighted variational lower bound and entropy regularization; (4) **SDAC** [30], which applies reweighted score matching without requiring optimal target samples; (5) **DIME** [19], which casts max-entropy policy improvement as approximate inference; and (6) **FlowRL** [15], which uses Wasserstein-2 regularization to guide flow policies toward high-value historical behaviors. Additional comparisons with tanh-squashed Gaussian SAC [20] and DSAC-T [21] are provided in Appendix E.

Implementation details. The detailed hyperparameters and reproducibility statement of other baselines are documented in Appendix B.2.

Main results. Representative training curves are presented in Figure 3, with the remaining curves deferred to Appendix C, and detailed numerical comparisons are listed in Table 1. GENIEO consistently delivers the best Total Average Return (TAR) across all 14 high-dimensional continuous control tasks among the generative-policy baselines in Table 1, based on observed mean returns.

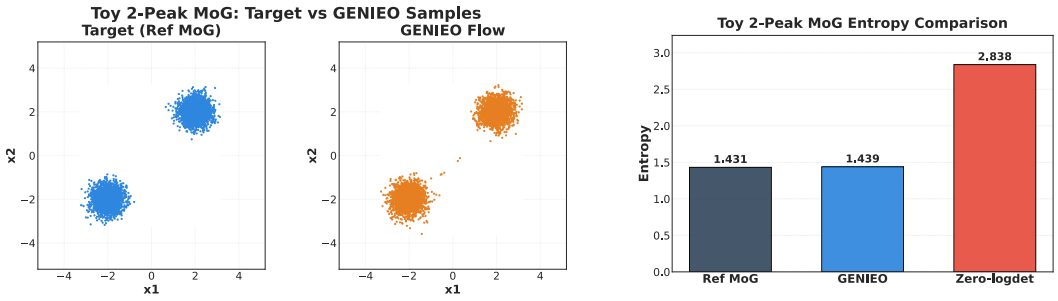
Results on the DMC Suite. Our GENIEO achieves an average TAR of **749**, outperforming the strongest generative baseline, FlowRL (641), by a substantial **+16.8%**, and exceeding DIME (586) by an even larger margin of **+27.8%**. Among the evaluated generative-policy baselines in Table 1, GENIEO achieves the highest observed mean return on all ten DMC tasks, including the highly demanding humanoid and dog tasks. In *Humanoid Walk*, GENIEO outperforms the strongest competing generative-policy baseline in observed mean return by **+5.0%** (730 vs. 695), and in the notoriously difficult *Dog Run*, it leads by a large **+23.0%** (476 vs. 387). Compared with direct Q-gradient methods such as DACER and QSM and other generative baselines such as FlowRL and DIME, GENIEO shows faster learning and stronger final performance on these high-dimensional control tasks.

Results on the HumanoidBench. GENIEO again achieves the best average performance on this highly non-linear, whole-body control suite with an average TAR of **1441**. This marks a substantial **+15.5%** improvement over the second-best SDAC (1248), and a strong **+22.8%** gain over FlowRL (1173). Among the evaluated generative-policy baselines in Table 1, GENIEO achieves the highest observed mean return on all four HumanoidBench tasks. For example, in the complex navigation task *H1 Sit Hard*, GENIEO improves over the second-best method by **+27.5%** (621 vs. 487), and in the delicate stabilization task *H1 Balance Simple*, it exceeds the second-best by **+26.0%** (184 vs. 146).

In summary, GENIEO shows consistent advantages across both benchmark suites. These results support the proposed tractable entropy-regularized formulation for off-policy generative policy learning.

4.2 Ablation Study

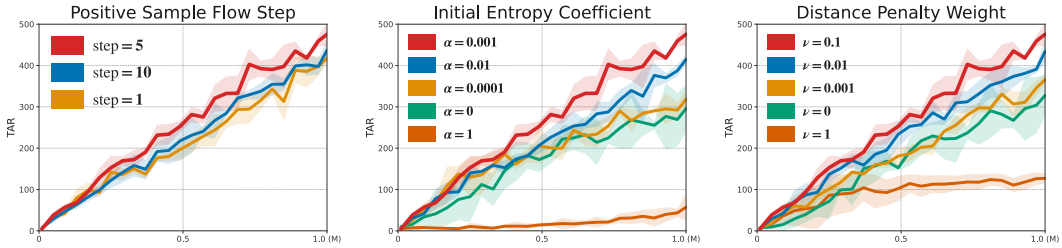
We first validate entropy computation in a toy setting, and then conduct three RL ablation studies to assess key design choices and hyperparameters. We select the *Dog-run* task in DMC Suite for all RL ablations due to its high-dimensional state-action spaces.



(a) Target and GENIEO samples

(b) Entropy comparison

Figure 4: **Toy entropy validation.** GENIEO reproduces the two-peak MoG structure and closely matches the reference entropy using the affine change-of-variables formula.



(a) Ablation study on the flow steps

(b) Ablation study on the initial entropy coefficient

(c) Ablation study on the distance penalty weight

Figure 5: **Ablation study curves.** We select the *Dog-run* task in DMC Suite with high dimensionality to perform all RL ablation experiments. The x-axis represents environment steps (in millions) and the y-axis represents the Total Average Return (TAR).

Toy entropy validation. We validate GENIEO's entropy computation on a two-peak mixture-of-Gaussians toy task. Figure 4 shows both sample fitting and entropy comparison, GENIEO closely matches the reference entropy (1.439 vs. 1.431), unlike the zero-logdet baseline (2.838). Appendix D provides details.

Flow steps. We study the effect of flow steps with $step \in \{1, 5, 10\}$. As shown in Figure 5a, $step = 5$ achieves the highest Total Average Return and fastest convergence. In contrast, $step = 1$ limits expressiveness, while $step = 10$ slightly degrades performance, likely due to more complicated optimization dynamics. Thus, $step = 5$ provides a favorable trade-off between representational capacity and learning efficiency.

Initial entropy coefficient. We study the initial entropy coefficient (α) over $\{0, 0.0001, 0.0010, 0.0100, 1.0000\}$. As shown in Figure 5b, $\alpha = 0.0010$ achieves the best final performance and stability. Using a zero initial coefficient ($\alpha = 0$) with automatic temperature adaptation lowers returns, while an overly large coefficient ($\alpha = 1.0000$) can dominate the reward signal and hinder directed optimization. Thus, $\alpha = 0.0010$ provides a suitable balance between policy stochasticity and stable policy improvement.

Coherence penalty weight. We evaluate the coherence penalty weight (ν) over $\{0, 0.0010, 0.0100, 0.1000, 1.0000\}$. As shown in Figure 5c, $\nu = 0.1000$ performs best by balancing policy improvement and coherence regularization. Removing the penalty ($\nu = 0$) weakens the coupling between paired dummy-action variables, while a large penalty ($\nu = 1.0000$) over-constrains the actor and slows learning. Thus, $\nu = 0.1000$ maintains coherent dual-variable generation while preserving flexibility for policy improvement.

5 Related Works

Generative Policies for RL. Diffusion models have become powerful tools for expressive distribution modeling, from DDPMs [13] and DDIMs [31] to score-based SDEs [32]. Flow-based generative models provide another route for continuous distribution transformation, including flow matching [14], rectified flows [33], stochastic interpolants [34], and mean flows [35]. These advances have inspired generative policies in RL: Diffuser formulates decision making as trajectory generation [36], while Diffusion-QL and IDQL use diffusion policies for expressive action modeling in offline RL [9, 11]. More recently, DIPO extends diffusion policy representations to online model-free RL [16], and FlowRL studies flow-based policies for online RL [15]. Together, these works highlight the potential of generative policies as expressive policy classes, while motivating closer study of their likelihood evaluation and entropy regularization.

Entropy Regularization and Likelihood Evaluation for Generative Policies. Maximum entropy RL augments the reward with policy entropy to encourage exploration, with SAC as a widely used off-policy actor-critic instantiation [23, 20]. KL-regularized methods such as MPO also constrain policy updates through relative-entropy objectives [37], and recent SAC-style variants further refine entropy-regularized continuous control [21, 3]. An unsquashed Gaussian has closed-form entropy, while a tanh-squashed Gaussian has tractable sample log-density and typically uses Monte Carlo entropy estimation. Diffusion and continuous-time flow policies often make likelihood evaluation difficult due to approximate reverse processes, numerical solvers, or intractable Jacobian terms. Recent generative policy-gradient methods study related likelihood or policy-ratio issues, including DPPO [10], FPO [38], and GenPO [39]. Closest to our work, GenPO also uses a doubled dummy-action construction to support exact likelihood evaluation and sample-based entropy and KL estimation for on-policy generative policy optimization [39]. GENIEO uses a discrete-time affine invertible dummy-action policy with a parameter-dependent log-determinant, so that the dummy-action entropy yields a non-degenerate optimization signal. GENIEO's contribution is to integrate this tractable entropy regularization into an off-policy actor-critic framework with replay-based soft Bellman learning.

6 Conclusion

In this paper, we introduced GENIEO, an off-policy generative RL framework that uses tractable entropy regularization of an augmented dummy-action policy for off-policy learning. By designing a dual-variable affine invertible flow, GENIEO enables tractable dummy-action likelihood and entropy evaluation without numerical ODE inversion, bridging expressive generative policy learning with maximum entropy off-policy RL. Experiments on DMControl and H-Bench show that GENIEO achieves higher observed suite-average returns than the evaluated diffusion- and flow-based generative RL baselines, suggesting that tractable dummy-action likelihood and entropy regularization is a promising direction for generative policy learning in high-dimensional control.

Limitation and Future Work. GENIEO doubles the latent dimensionality and defines entropy in the augmented dummy-action space, which may pose scaling challenges in extremely high-dimensional action spaces. It also requires careful tuning of the initial entropy coefficient α and coherence penalty weight ν. Future work will study projected environment-action entropy and temporally extended novelty-seeking mechanisms for extremely sparse rewards. Precision-sensitive tasks may additionally require lower execution-time stochasticity.

Acknowledgments and Disclosure of Funding

This work was supported by the National Science and Technology Major Project (No. 2025ZD1606200), the Tsinghua-Efort Joint Research Center for EAI Computation and Perception, and SunRising AI Lab. The authors declare no competing interests.

References

[1] Guojian Zhan, Xiangteng Zhang, Feihong Zhang, Letian Tao, and Shengbo Eben Li. Bicriteria policy optimization for high-accuracy reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.

[2] Yao Lyu, Xiangteng Zhang, Shengbo Eben Li, Jingliang Duan, Letian Tao, Qing Xu, Lei He, and Keqiang Li. Conformal symplectic optimization for stable reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(6):11049–11063, 2024.

[3] Guojian Zhan, Likun Wang, Pengcheng Wang, Feihong Zhang, Jingliang Duan, Masayoshi Tomizuka, and Shengbo Eben Li. Mind your entropy: From maximum entropy to trajectory entropy-constrained rl. *arXiv preprint arXiv:2511.11592*, 2025.

[4] Yinuo Wang, Likun Wang, Mining Tan, Wenjun Zou, Xujie Song, Wenxuan Wang, Tong Liu, Guojian Zhan, Tianze Zhu, Shiqi Liu, et al. Enhanced dacer algorithm with high diffusion efficiency. *arXiv preprint arXiv:2505.23426*, 2025.

[5] Guojian Zhan, Likun Wang, Feihong Zhang, Yang Guan, and Shengbo Eben Li. Harmonized dual policy improvement for modelic reinforcement learning. In *Forty-third International Conference on Machine Learning*.

[6] Tianyi Zhang, Likun Wang, Guojian Zhan, Feihong Zhang, Yang Guan, Yao Lyu, and Shengbo Eben Li. Langevin rollout optimization for modelic reinforcement learning. In *Forty-third International Conference on Machine Learning*.

[7] Feihong Zhang, Guojian Zhan, Bin Shuai, Tianyi Zhang, Jingliang Duan, and Shengbo Eben Li. Distributional soft actor-critic with harmonic gradient for safe and efficient autonomous driving in multi-lane scenarios. In *2025 IEEE Intelligent Vehicles Symposium (IV)*, pages 1964–1970. IEEE, 2025.

[8] Tianze Zhu, Yinuo Wang, Wenjun Zou, Tianyi Zhang, Likun Wang, Letian Tao, Feihong Zhang, Yao Lyu, and Shengbo Eben Li. Real-time generative policy via langevin-guided flow matching for autonomous driving. *arXiv preprint arXiv:2603.02613*, 2026.

[9] Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. *arXiv preprint arXiv:2208.06193*, 2022.

[10] Allen Z Ren, Justin Lidard, Lars L Ankile, Anthony Simeonov, Pulkit Agrawal, Anirudha Majumdar, Benjamin Burchfiel, Hongkai Dai, and Max Simchowitz. Diffusion policy policy optimization. *arXiv preprint arXiv:2409.00588*, 2024.

[11] Philippe Hansen-Estruch, Ilya Kostrikov, Michael Janner, Jakub Grudzien Kuba, and Sergey Levine. Idql: Implicit q-learning as an actor-critic method with diffusion policies. *arXiv preprint arXiv:2304.10573*, 2023.

[12] Guojian Zhan, Letian Tao, Pengcheng Wang, Yixiao Wang, Yiheng Li, Yuxin Chen, Masayoshi Tomizuka, and Shengbo Eben Li. Mean flow policy with instantaneous velocity constraint for one-step action generation. *arXiv preprint arXiv:2602.13810*, 2026.

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[14] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

[15] Lei Lv, Yunfei Li, Yu Luo, Fuchun Sun, Tao Kong, Jiafeng Xu, and Xiao Ma. Flow-based policy for online reinforcement learning. *arXiv preprint arXiv:2506.12811*, 2025.

[16] Long Yang, Zhixiong Huang, Fenghao Lei, Yucun Zhong, Yiming Yang, Cong Fang, Shiting Wen, Binbin Zhou, and Zhouchen Lin. Policy representation via diffusion probability model for reinforcement learning. *arXiv preprint arXiv:2305.13122*, 2023.

[17] Yinuo Wang, Likun Wang, Yuxuan Jiang, Wenjun Zou, Tong Liu, Xujie Song, Wenxuan Wang, Liming Xiao, Jiang Wu, Jingliang Duan, et al. Diffusion actor-critic with entropy regulator. *Advances in Neural Information Processing Systems*, 37:54183–54204, 2024.

[18] Shutong Ding, Ke Hu, Zhenhao Zhang, Kan Ren, Weinan Zhang, Jingyi Yu, Jingya Wang, and Ye Shi. Diffusion-based reinforcement learning via q-weighted variational policy optimization. *Advances in Neural Information Processing Systems*, 37:53945–53968, 2024.

[19] Onur Celik, Zechu Li, Denis Blessing, Ge Li, Daniel Palenicek, Jan Peters, Georgia Chalvatzaki, and Gerhard Neumann. Dime: Diffusion-based maximum entropy reinforcement learning. *arXiv preprint arXiv:2502.02316*, 2025.

[20] 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.

[21] Jingliang Duan, Wenxuan Wang, Liming Xiao, Jiaxin Gao, Shengbo Eben Li, Chang Liu, Ya-Qin Zhang, Bo Cheng, and Keqiang Li. Distributional soft actor-critic with three refinements. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[22] Shengbo Eben Li. Reinforcement learning for sequential decision and optimal control. 2023.

[23] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pages 1352–1361. PMLR, 2017.

[24] Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pages 1530–1538. PMLR, 2015.

[25] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[26] Joseph K Blitzstein and Jessica Hwang. *Introduction to probability*. Chapman and Hall/CRC, 2019.

[27] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

[28] Carmelo Sferrazza, Dun-Ming Huang, Xingyu Lin, Youngwoon Lee, and Pieter Abbeel. Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation. *arXiv preprint arXiv:2403.10506*, 2024.

[29] Michael Psenka, Alejandro Escontrela, Pieter Abbeel, and Yi Ma. Learning a diffusion model policy from rewards via q-score matching. *arXiv preprint arXiv:2312.11752*, 2023.

[30] Haitong Ma, Tianyi Chen, Kai Wang, Na Li, and Bo Dai. Efficient online reinforcement learning for diffusion policy. *arXiv preprint arXiv:2502.00361*, 2025.

[31] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[32] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[33] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[34] Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

[35] Zhengyang Geng, Mingyang Deng, Xingjian Bai, J Zico Kolter, and Kaiming He. Mean flows for one-step generative modeling. *arXiv preprint arXiv:2505.13447*, 2025.

[36] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022.

[37] Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. *arXiv preprint arXiv:1806.06920*, 2018.

[38] David McAllister, Songwei Ge, Brent Yi, Chung Min Kim, Ethan Weber, Hongsuk Choi, Haiwen Feng, and Angjoo Kanazawa. Flow matching policy gradients. *arXiv preprint arXiv:2507.21053*, 2025.

[39] Shutong Ding, Ke Hu, Shan Zhong, Haoyang Luo, Weinan Zhang, Jingya Wang, Jun Wang, and Ye Shi. Genpo: Generative diffusion models meet on-policy reinforcement learning. *arXiv preprint arXiv:2505.18763*, 2025.

A Theoretical Analysis

A.1 Proof of Change-of-Variables Lemma

Proof. Since f is a C^1 diffeomorphism, both f and f^{-1} are continuously differentiable. Denote the Jacobian matrix of f at x by

$$J_f(x) = \frac{\partial f(x)}{\partial x}.$$

Similarly, denote the Jacobian matrix of f^{-1} at y by

$$J_{f^{-1}}(y) = \frac{\partial f^{-1}(y)}{\partial y}.$$

Let X be a random variable with density $q(x)$, and define

$$Y = f(X).$$

We aim to derive the density $p(y)$ of Y.

For any measurable set $A \subseteq \mathbb{R}^n$, we have

$$\mathbb{P}(Y \in A) = \mathbb{P}(f(X) \in A).$$

Since f is bijective, the event $\{f(X) \in A\}$ is equivalent to the event $\{X \in f^{-1}(A)\}$. Therefore,

$$\mathbb{P}(Y \in A) = \mathbb{P}(X \in f^{-1}(A)).$$

Because X has density $q(x)$, this probability can be written as

$$\mathbb{P}(X \in f^{-1}(A)) = \int_{f^{-1}(A)} q(x)\, dx.$$

We now apply the multivariable change-of-variables theorem with

$$y = f(x), \qquad x = f^{-1}(y).$$

The volume element transforms as

$$dx = \left|\det J_{f^{-1}}(y)\right| \, dy.$$

Thus,

$$\int_{f^{-1}(A)} q(x)\, dx = \int_A q(f^{-1}(y)) \left|\det J_{f^{-1}}(y)\right| \, dy.$$

Combining the above equations gives

$$\mathbb{P}(Y \in A) = \int_A q(f^{-1}(y)) \left|\det J_{f^{-1}}(y)\right| \, dy.$$

Since this equality holds for every measurable set A, a density of Y is given by

$$p(y) = q(f^{-1}(y)) \left|\det J_{f^{-1}}(y)\right|.$$

It remains to express the inverse Jacobian in terms of the forward mapping f. Since $x = f^{-1}(y)$, by the inverse function theorem,

$$J_{f^{-1}}(y) = J_f(f^{-1}(y))^{-1}.$$

Taking determinants on both sides yields

$$\det J_{f^{-1}}(y) = \left(\det J_f(f^{-1}(y))\right)^{-1}.$$

Therefore,

$$\left|\det J_{f^{-1}}(y)\right| = \left|\det J_f(x)\right|^{-1}, \qquad x = f^{-1}(y).$$

Substituting this into the expression for $p(y)$ gives

$$p(y) = q(x) \left|\det J_f(x)\right|^{-1}, \qquad x = f^{-1}(y).$$

Using $J_f(x) = \frac{\partial f(x)}{\partial x}$, we obtain

$$p(y) = q(x) \left|\det \frac{\partial f(x)}{\partial x}\right|^{-1}, \qquad x = f^{-1}(y).$$

This completes the proof. □

A.2 Proof of Exact Invertibility of GENIEO Affine Alternating Update

Proof. Fix a state s, a flow time t, and a step size $h \in \mathbb{R}$. For notational simplicity, define

$$u(\cdot) = u_\theta(\cdot, t, s), \qquad \ell(\cdot) = \ell_\theta(\cdot, t, s),$$

and

$$w(\cdot) = w_\theta(\cdot, t, s), \qquad \rho(\cdot) = \rho_\theta(\cdot, t, s).$$

By assumption,

$$u, \ell, w, \rho : \mathbb{R}^d \to \mathbb{R}^d$$

are continuously differentiable. All exponentials and products below are applied element-wise.

The one-step GENIEO affine alternating update defines a map

$$F_h : \mathbb{R}^{2d} \to \mathbb{R}^{2d}, \qquad F_h(x, y) = (X, Y),$$

where

$$X = x \odot \exp(h\ell(y)) + h\, u(y),$$

and

$$Y = y \odot \exp(h\rho(X)) + h\, w(X).$$

Since the exponential function is strictly positive component-wise, the scaling vectors

$$\exp(h\ell(y)), \qquad \exp(h\rho(X))$$

have no zero entries. Therefore, the element-wise divisions required for inversion are well-defined.

We prove exact invertibility by explicitly constructing the inverse map. Given arbitrary transformed variables $(X, Y) \in \mathbb{R}^{2d}$, define

$$y = (Y - h\, w(X)) \odot \exp(-h\rho(X)),$$

and then define

$$x = (X - h\, u(y)) \odot \exp(-h\ell(y)).$$

This defines a candidate inverse map

$$G_h : \mathbb{R}^{2d} \to \mathbb{R}^{2d}, \qquad G_h(X, Y) = (x, y).$$

Equivalently,

$$G_h(X, Y) = \left((X - h\, u(y)) \odot \exp(-h\ell(y)),\ y\right),$$

where

$$y = (Y - h\, w(X)) \odot \exp(-h\rho(X)).$$

We now verify that G_h is indeed the inverse of F_h. First, take any $(x, y) \in \mathbb{R}^{2d}$ and let

$$(X, Y) = F_h(x, y).$$

By the definition of F_h,

$$X = x \odot \exp(h\ell(y)) + h\, u(y),$$

and

$$Y = y \odot \exp(h\rho(X)) + h\, w(X).$$

Applying G_h to (X, Y), the recovered second component is

$$\begin{aligned} y' &= (Y - h\, w(X)) \odot \exp(-h\rho(X)) \\ &= (y \odot \exp(h\rho(X)) + h\, w(X) - h\, w(X)) \odot \exp(-h\rho(X)) \\ &= y \odot \exp(h\rho(X)) \odot \exp(-h\rho(X)) \\ &= y. \end{aligned}$$

Using $y' = y$, the recovered first component is

$$\begin{aligned} x' &= (X - h\, u(y')) \odot \exp(-h\ell(y')) \\ &= (X - h\, u(y)) \odot \exp(-h\ell(y)) \\ &= (x \odot \exp(h\ell(y)) + h\, u(y) - h\, u(y)) \odot \exp(-h\ell(y)) \\ &= x \odot \exp(h\ell(y)) \odot \exp(-h\ell(y)) \\ &= x. \end{aligned}$$

Therefore,

$$G_h(F_h(x,y)) = (x,y)$$

for every $(x,y) \in \mathbb{R}^{2d}$.

Conversely, take any $(X,Y) \in \mathbb{R}^{2d}$ and let

$$(x,y) = G_h(X,Y).$$

By construction,

$$y = (Y - h\,w(X)) \odot \exp(-h\rho(X)),$$

and

$$x = (X - h\,u(y)) \odot \exp(-h\ell(y)).$$

Applying F_h to (x,y), the first transformed component is

$$\begin{aligned} X' &= x \odot \exp(h\ell(y)) + h\,u(y) \\ &= (X - h\,u(y)) \odot \exp(-h\ell(y)) \odot \exp(h\ell(y)) + h\,u(y) \\ &= X - h\,u(y) + h\,u(y) \\ &= X. \end{aligned}$$

The second transformed component is

$$\begin{aligned} Y' &= y \odot \exp(h\rho(X')) + h\,w(X') \\ &= y \odot \exp(h\rho(X)) + h\,w(X) \\ &= (Y - h\,w(X)) \odot \exp(-h\rho(X)) \odot \exp(h\rho(X)) + h\,w(X) \\ &= Y - h\,w(X) + h\,w(X) \\ &= Y. \end{aligned}$$

Thus,

$$F_h(G_h(X,Y)) = (X,Y)$$

for every $(X,Y) \in \mathbb{R}^{2d}$.

Since both compositions satisfy

$$G_h \circ F_h = \mathrm{Id}_{\mathbb{R}^{2d}}, \qquad F_h \circ G_h = \mathrm{Id}_{\mathbb{R}^{2d}},$$

the map F_h is bijective and $G_h = F_h^{-1}$. Moreover, because u, ℓ, w, ρ are continuously differentiable and the exponential function is smooth, both F_h and its explicit inverse G_h are continuously differentiable. Hence F_h is a C^1 diffeomorphism.

It remains to compute the log-determinant of F_h. We decompose F_h into two triangular transformations. The first transformation maps

$$(x,y) \mapsto (X,y), \qquad X = x \odot \exp(h\ell(y)) + h\,u(y).$$

Its Jacobian matrix has the block-triangular form

$$\frac{\partial(X,y)}{\partial(x,y)} = \begin{pmatrix} D_x & * \\ 0 & I \end{pmatrix},$$

where

$$D_x = \mathrm{diag}(\exp(h\ell(y))).$$

Therefore,

$$\det \frac{\partial(X,y)}{\partial(x,y)} = \det(D_x) = \prod_{j=1}^{d} \exp(h\ell_j(y)) = \exp\big(h\,\mathbf{1}^\top \ell(y)\big).$$

The second transformation maps

$$(X,y) \mapsto (X,Y), \qquad Y = y \odot \exp(h\rho(X)) + h\,w(X).$$

Its Jacobian matrix has the block-triangular form

$$\frac{\partial(X,Y)}{\partial(X,y)} = \begin{pmatrix} I & 0 \\ * & D_y \end{pmatrix},$$

where

$$D_y = \operatorname{diag}(\exp(h\rho(X)))\,.$$

Thus,

$$\det \frac{\partial(X,Y)}{\partial(X,y)} = \det(D_y) = \prod_{j=1}^{d} \exp(h\rho_j(X)) = \exp\big(h\,\mathbf{1}^\top \rho(X)\big)\,.$$

Since F_h is the composition of these two transformations, the chain rule for Jacobians gives

$$\det \frac{\partial F_h(x,y)}{\partial(x,y)} = \exp\big(h\,\mathbf{1}^\top \ell(y)\big)\exp\big(h\,\mathbf{1}^\top \rho(X)\big)\,.$$

Because both exponential factors are positive, the absolute value does not change the determinant sign. Therefore,

$$\begin{aligned}\log\left|\det \frac{\partial F_h(x,y)}{\partial(x,y)}\right| &= \log\left[\exp\big(h\,\mathbf{1}^\top \ell(y)\big)\exp\big(h\,\mathbf{1}^\top \rho(X)\big)\right] \\ &= h\,\mathbf{1}^\top \ell(y) + h\,\mathbf{1}^\top \rho(X).\end{aligned}$$

Substituting back

$$\ell(\cdot) = \ell_\theta(\cdot, t, s), \qquad \rho(\cdot) = \rho_\theta(\cdot, t, s),$$

we obtain

$$\log\left|\det \frac{\partial F_h(x,y)}{\partial(x,y)}\right| = h\,\mathbf{1}^\top \ell_\theta(y,t,s) + h\,\mathbf{1}^\top \rho_\theta(X,t,s).$$

This completes the proof. □

A.3 Proof of Entropy Expression and Gradient

Proof. Fix a state s. For notational clarity, define

$$J_\theta(z;s) = \frac{\partial F_\theta(z;s)}{\partial z}.$$

By Theorem 2, each GENIEO affine alternating update is a C^1 diffeomorphism. Since a finite composition of C^1 diffeomorphisms is still a C^1 diffeomorphism, the full GENIEO transformation

$$F_\theta(\cdot;s) : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$$

is also a C^1 diffeomorphism. Moreover, by the chain rule for Jacobians, the log-determinant of the full transformation is the sum of the per-step log-determinants given in Eq. 13.

We assume the standard regularity conditions required for the following derivation: $F_\theta(\cdot;s)$ is differentiable with respect to θ, $\det J_\theta(z;s) \neq 0$ for all z in the domain of interest, and

$$\log|\det J_\theta(z;s)|$$

together with its θ-gradient is integrable under p_0, so that differentiation and expectation can be interchanged.

GENIEO generates the augmented dummy action by

$$\tilde{a} = F_\theta(z;s), \qquad z = (x_0, y_0) \sim p_0(z),$$

where p_0 is fixed and independent of θ. Since $F_\theta(\cdot;s)$ is a diffeomorphism, Lemma 1 gives the induced density of $\tilde{a}$:

$$\pi_\theta(\tilde{a} \mid s) = p_0(z)\,|\det J_\theta(z;s)|^{-1}\,, \qquad z = F_\theta^{-1}(\tilde{a};s).$$

Taking logarithms gives

$$\log \pi_\theta(\tilde{a} \mid s) = \log p_0(z) - \log|\det J_\theta(z;s)|\,, \qquad z = F_\theta^{-1}(\tilde{a};s).$$

By definition, the conditional entropy of the augmented dummy-action policy is

$$\mathcal{H}_\theta(s) = -\mathbb{E}_{\tilde{a}\sim\pi_\theta(\cdot|s)}\left[\log \pi_\theta(\tilde{a} \mid s)\right].$$

Since $\tilde{a} = F_\theta(z; s)$ with $z \sim p_0(z)$, the expectation over $\tilde{a}$ can be rewritten as an expectation over the base variable z:

$$\mathcal{H}_\theta(s) = -\mathbb{E}_{z\sim p_0}\left[\log \pi_\theta(F_\theta(z; s) \mid s)\right].$$

For $\tilde{a} = F_\theta(z; s)$, the inverse satisfies

$$F_\theta^{-1}(\tilde{a}; s) = F_\theta^{-1}(F_\theta(z; s); s) = z.$$

Therefore,

$$\log \pi_\theta(F_\theta(z; s) \mid s) = \log p_0(z) - \log\left|\det J_\theta(z; s)\right|.$$

Substituting this expression into the entropy gives

$$\mathcal{H}_\theta(s) = -\mathbb{E}_{z\sim p_0}\left[\log p_0(z) - \log\left|\det J_\theta(z; s)\right|\right].$$

Expanding the expectation, we obtain

$$\mathcal{H}_\theta(s) = -\mathbb{E}_{z\sim p_0}\left[\log p_0(z)\right] + \mathbb{E}_{z\sim p_0}\left[\log\left|\det J_\theta(z; s)\right|\right].$$

Since the base entropy is

$$\mathcal{H}(p_0) = -\mathbb{E}_{z\sim p_0}\left[\log p_0(z)\right],$$

we have

$$\mathcal{H}_\theta(s) = \mathcal{H}(p_0) + \mathbb{E}_{z\sim p_0}\left[\log\left|\det J_\theta(z; s)\right|\right].$$

Substituting back

$$J_\theta(z; s) = \frac{\partial F_\theta(z; s)}{\partial z},$$

we obtain

$$\mathcal{H}_\theta(s) = \mathcal{H}(p_0) + \mathbb{E}_{z\sim p_0}\left[\log\left|\det \frac{\partial F_\theta(z; s)}{\partial z}\right|\right].$$

We now derive the gradient with respect to θ. Since p_0 is independent of θ, the base entropy $\mathcal{H}(p_0)$ is constant with respect to θ. Hence,

$$\nabla_\theta \mathcal{H}_\theta(s) = \nabla_\theta \mathbb{E}_{z\sim p_0}\left[\log\left|\det J_\theta(z; s)\right|\right].$$

By the stated regularity conditions, differentiation can be exchanged with the expectation:

$$\nabla_\theta \mathcal{H}_\theta(s) = \mathbb{E}_{z\sim p_0}\left[\nabla_\theta \log\left|\det J_\theta(z; s)\right|\right].$$

Finally, substituting back the definition of $J_\theta(z; s)$ yields

$$\nabla_\theta \mathcal{H}_\theta(s) = \mathbb{E}_{z\sim p_0}\left[\nabla_\theta \log\left|\det \frac{\partial F_\theta(z; s)}{\partial z}\right|\right].$$

This completes the proof. □

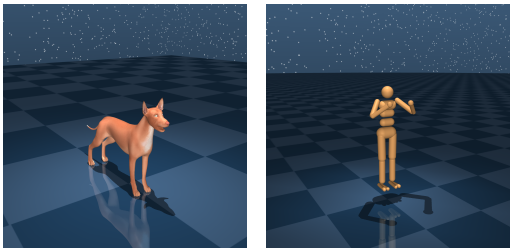
B Environmental Details

B.1 Benchmark Introduction

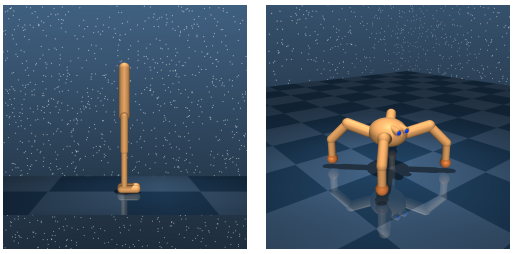
DeepMind Control Suite. We conduct our evaluation across ten demanding continuous control environments utilizing four distinct agent morphologies: `dog`, `humanoid`, `walker`, and `quadruped` (refer to Figure 6). These evaluated tasks are broadly classified into two primary groups: (1) *Standing tasks*, wherein the agent is expected to preserve an upright balanced posture (e.g., Stand), and (2) *Moving tasks*, which introduce the extra requirement that the agent travel at a designated target speed (e.g., Walk, Trot, Run).

For the moving scenarios, the overall reward is typically formulated by multiplying the standing stability reward with the forward velocity reward, specifically: ***Reward*** $=$ (***Standing reward***) $\times$ (***Forward velocity reward***).

HumanoidBench. We incorporate four intricate whole-body locomotion and manipulation benchmarks utilizing the Unitree H1 robot. The robot is consistently initialized in a standing stance, with stochastic noise injected into all joint configurations during every episode reset. The specific objectives and associated reward mechanisms are detailed below.



(a) Dog (b) Humanoid



(c) Walker (d) Quadruped

Figure 6: Four distinct agent morphologies assessed within the DeepMind Control Suite.

Dog: A high-dimensional four-legged agent engineered for intricate locomotion, encompassing tasks such as stand, walk, trot, and run.

Humanoid: An advanced 3D bipedal agent possessing 21 degrees of freedom, demanding sophisticated equilibrium and movement coordination.

Walker: A planar two-legged agent constrained to 2D movement, acting as a foundational benchmark to study essential locomotion dynamics.

Quadruped: A versatile 3D four-legged agent that requires the synchronization of multiple limbs to produce stable walking gaits.

Rewards: Within moving tasks, the final reward is usually computed as the product of a ***Standing reward*** (which encourages an upright and stable posture) and a ***Forward velocity reward*** (which ensures the agent progresses at the desired target speed).

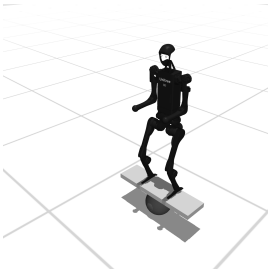


Figure 7: H1 Balance Simple

Objective. Maintain balance upon an unstable board. Within the `simple` configuration, the spherical pivot located underneath the board is kept strictly fixed.

Reward: $R(s,a) = (e \times \texttt{still}) \times (\texttt{height}_{\text{robot}} \times \texttt{upright})$, where `still` promotes stationary equilibrium, e incentivizes minimal control effort, and the other components reward the preservation of a high and vertical torso.

Termination. 1000 steps, or early termination when $z_{\text{robot}} < 0.8$, the board strikes the floor surface, or the pivot sphere contacts any object apart from the floor and the board itself.

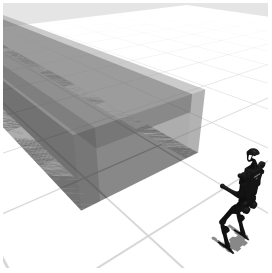


Figure 8: H1 Crawl

Objective. Sustain a forward velocity approximating $1\,\mathrm{m/s}$ while successfully navigating through the interior of a tunnel.

Reward: $R = \texttt{tunnel} \times \big(0.1 + 0.25\min(\texttt{height}_{\text{crawl}}, \texttt{height}_{\text{IMU}}) + 0.25\texttt{orientation} + 0.4\texttt{speed}\big)$. In this formulation, `tunnel` rewards remaining within the lateral boundaries of the tunnel, `orientation` aligns the robot body with a defined crawling quaternion, and the `height` variables encourage a low and stable crawling stance.

Termination. 1000 continuous steps.

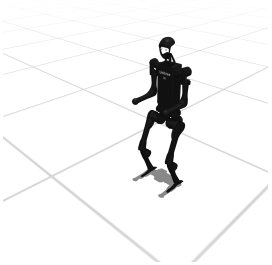


Figure 9: H1 Reach

Objective. Extend the left hand to touch a randomly generated three-dimensional target point in space.

Reward: $R(s,a) = 10 - 4\texttt{penalty}_{\text{motion}} + \texttt{health} + \texttt{close} + \texttt{success}$. The `health` component $(5 - 2z_{\text{pelvis.proj}})$ rewards the agent for keeping its pelvis elevated, whereas `close` and `success` grant tiered bonuses as the hand moves within specified distance thresholds toward the target.

Termination. 1000 continuous steps.

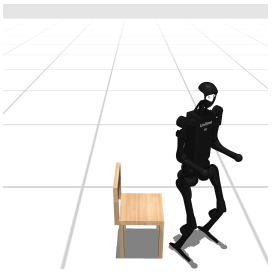


Figure 10: H1 Sit Hard

Objective. Successfully sit upon a provided chair. Under the `hard` configuration, the robot's initial position and orientation $\alpha \in [-1.8, 1.8]$ rad are randomized, necessitating active navigation and precise alignment prior to the sitting action.

Reward: $R(s,a) = ((0.5 \cdot \texttt{sitting_z} + 0.5 \cdot \texttt{sitting_x} \cdot \texttt{sitting_y}) \times \texttt{upright} \times \texttt{posture}) \times e \times \texttt{mean}(\texttt{still_x}, \texttt{still_y})$. This calculation focuses on minimizing the spatial gap between the robot's pelvis and the seat of the chair, while simultaneously enforcing a structurally proper seated posture.

Termination. 1000 steps, or early termination if $z_{\text{pelvis}} < 0.5$.

B.2 Reproducibility Statement & Detailed Hyperparameters

We base our foundational experimental framework on the released official FlowRL codebase (https://github.com/bytedance/FlowRL). To facilitate reproducible research, we provide the core implementation of GENIEO and an accompanying README in the camera-ready code archive. The archive documents the main algorithmic components and dependencies, while the experimental settings and hyperparameters are reported below. During the training and evaluation phases, we utilize Weights & Biases (WandB) for consistent experimental tracking, metric logging, and transparent visualization across all random seeds.

In this paper, we evaluate each algorithm over five random seeds and report the mean and standard deviation to characterize performance variability across runs. The computational infrastructure utilized for our experiments consists of high-performance computation nodes equipped with 4 $\times$ NVIDIA RTX 4090 GPUs. Taking the highly demanding *Dog Run* task in the DMC Suite as a practical example, the computational time required to train for 1M iterations is approximately 10 hours.

For baseline comparisons, we implemented the respective algorithms directly within the FlowRL framework or adopted their official repository implementations to ensure the utmost fairness. Specifically, we adapt the official codebases for DACER (https://github.com/happy-yan/DACER-Diffusion-with-Online-RL), SDAC (https://github.com/mahaitongdae/diffusion_policy_online_rl), QSM (https://github.com/escontra/score_matching_rl), QVPO (https://github.com/wadx2019/qvpo/), and DIME (https://github.com/ALRhub/DIME). It is worth noting that the initial hyperparameter settings of some selected baseline algorithms differ from our codebase, such as DIME algorithm. To ensure a strictly fair comparison, we aligned its parameter settings with our unified framework. Overall, we strictly align critical experimental settings across all algorithms.

We use the DeepMind Control Suite [27] and HumanoidBench [28] according to their official licenses, and we use FlowRL and baseline repositories under their respective open-source licenses. The corresponding repository URLs are listed above.

The relevant hyperparameter configurations for all evaluated algorithms are summarized in Table 2. These values are kept consistent across all evaluated tasks unless otherwise specified in the task-specific sections.

Table 2: Comprehensive hyperparameter configurations for GENIEO and baseline algorithms across evaluated benchmark suites.

Hyperparameter	DACER	SDAC	QSM	QVPO	DIME	FlowRL	GENIEO (Ours)
General RL Settings							
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam	Adam
Update-to-data ratio	1	1	1	1	1	1	1
Discount factor (γ)	0.99	0.99	0.99	0.99	0.99	0.99	0.99
Batch size	256	256	256	256	256	256	256
Buffer size	10^6	10^6	10^6	10^6	10^6	10^6	10^6
Exploration steps	5000	5000	5000	5000	5000	5000	5000
Network Architecture & Optimization							
Critic hidden depth	3	3	2	2	2	4	4
Critic hidden size	512	512	512	512	512	512	512
Actor/Score hidden depth	3	3	3	2	3	2	2
Actor/Score hidden size	256	512	256	256	256	512	512
Learning rate (Critic)	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}
Learning rate (Actor/Score)	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-4}	3×10^{-5}
Learning rate (Temperature)	3×10^{-2}	3×10^{-4}	N/A	N/A	1×10^{-3}	N/A	3×10^{-5}
Algorithm-Specific & Generative Mechanics							
Target entropy ($\bar{\mathcal{H}}$)	$-2\dim(\mathcal{A})$	$-2\dim(\mathcal{A})$	N/A	N/A	$-2\dim(\mathcal{A})$	N/A	$-2\dim(\mathcal{A})$
Num. Bins / Quantiles	2	N/A	N/A	N/A	100	0.9	0.9
Generative flow/diffusion steps	20	20	15	20	16	1	5
Prior distribution $\mathcal{N}(0, \sigma^2\mathbf{I})$	$\sigma = 1.0$	$\sigma = 1.0$	$\sigma = 1.0$	$\sigma = 1.0$	$\sigma = 1.0$	$\sigma = 1.0$	$\sigma = 1.0$
Score-Q align. factor	N/A	N/A	50	N/A	N/A	N/A	N/A
GENIEO Exclusive Hyperparameters							
Coherence weight (ν)	N/A	N/A	N/A	N/A	N/A	N/A	0.1
Initial entropy coef (α_0)	N/A	N/A	N/A	N/A	N/A	N/A	0.001

C Additional Curves

Figure 11 reports the remaining task-level training curves that are not shown in the main text due to space limitations. Together with Figure 3, these curves cover all evaluated DMControl and HumanoidBench tasks.

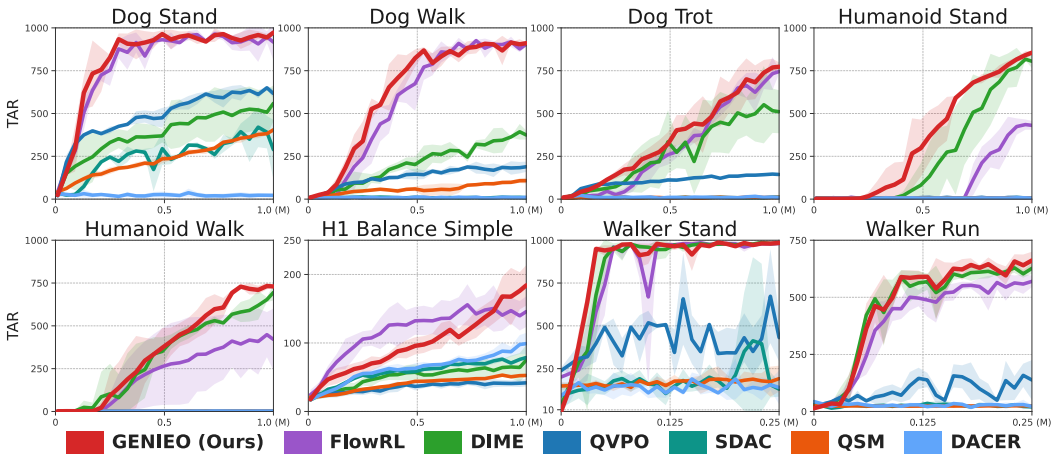


Figure 11: **Additional performance curves on DMControl and HumanoidBench.** Solid lines show mean episode returns over 5 random seeds, and shaded bands denote ± 1 standard deviation.

D Toy Entropy Validation

To further validate the entropy computation mechanism of GENIEO, we conduct a controlled toy experiment on a two-dimensional two-peak mixture-of-Gaussians (MoG) target distribution. The target distribution consists of two equally weighted Gaussian components centered at $(-2, -2)$ and $(2, 2)$, each with covariance $\sigma^2 I$ and $\sigma = 0.35$. Since the target density is explicitly available, we compute a high-precision reference entropy by numerical integration over a 1200×1200 grid on $[-5.5, 5.5]^2$. The resulting probability mass check is 1.0000, indicating that the grid integration provides a reliable reference.

We then train a GENIEO affine flow to fit this target distribution by maximum likelihood. The base distribution is a two-dimensional standard Gaussian $p_0(z) = \mathcal{N}(0, I_2)$, whose entropy is $\mathcal{H}(p_0) = 2.8379$. The GENIEO toy flow uses 8 affine alternating steps with log-scale bound 1.0, and is trained for 30k steps with Adam. After training, the entropy of the learned flow is estimated by the change-of-variables formula:

$$\widehat{\mathcal{H}}_{\mathrm{GENIEO}} = \mathcal{H}(p_0) + \mathbb{E}_{z\sim p_0}\left[\log\left|\det\frac{\partial F_\theta(z)}{\partial z}\right|\right], \tag{19}$$

where the expectation is approximated using 200k base samples. We also compare against a zero-logdet baseline, which ignores the affine log-determinant and therefore reduces to the base entropy.

As shown in Table 3, the GENIEO flow entropy is 1.4387, closely matching the high-precision reference entropy 1.4314 with an absolute error of only 0.0074. In contrast, the zero-logdet baseline gives an entropy of 2.8379, resulting in a much larger absolute error of 1.4065. Figure 4 visualizes this comparison, showing that the affine log-determinant is essential for capturing the entropy change induced by the learned transformation.

Table 3: Toy entropy validation on a two-peak MoG target. GENIEO closely matches the high-precision reference entropy, while the zero-logdet baseline fails to capture the entropy change induced by the affine transformation.

Method	**Entropy**	**Abs. Error**	**Rel. Error**
Reference MoG	1.4314	–	–
GENIEO flow entropy	1.4387	0.0074	0.51%
Zero-logdet baseline	2.8379	1.4065	98.26%

The learned GENIEO flow also achieves a small NLL gap of 0.0071 on target samples, indicating that the fitted flow distribution is close to the target distribution. This is visually supported by Figure 4, where the generated GENIEO samples reproduce the two separated modes of the target MoG distribution. Moreover, the learned mean log-determinant is -1.3991, close to the required entropy difference $\mathcal{H}_{\mathrm{target}} - \mathcal{H}(p_0) = 1.4314 - 2.8379 = -1.4065$. Together, these results confirm that the affine log-determinant provides a non-degenerate and accurate entropy estimate.

E Additional Comparisons with SAC and DSAC-T

We provide additional comparisons with SAC and DSAC-T to complement the generative-policy baselines evaluated in the main text.

GENIEO achieves higher observed suite-average returns on both benchmarks. SAC has higher observed mean returns on Quadruped Walk and H1 Reach, indicating that the advantage is not uniform across tasks.

Table 4: Total Average Return (TAR) on 10 DMC Suite tasks and 4 H-Bench tasks. Mean ± standard deviation over five seeds. Bold denotes the highest observed mean and does not imply statistical significance. Higher is better.

Task	**SAC**	**DSAC-T**	**GENIEO (Ours)**
DMC Suite			
Dog Stand	732 ± 101	282 ± 86	**973 ± 10**
Dog Walk	98 ± 143	81 ± 14	**911 ± 5**
Dog Run	102 ± 96	120 ± 65	**476 ± 28**
Dog Trot	442 ± 175	39 ± 16	**773 ± 25**
Humanoid Stand	568 ± 23	418 ± 390	**855 ± 10**
Humanoid Walk	522 ± 41	457 ± 18	**730 ± 6**
Humanoid Run	167 ± 23	70 ± 59	**210 ± 13**
Quadruped Walk	**947 ± 11**	720 ± 168	913 ± 31
Walker Stand	982 ± 3	958 ± 13	**984 ± 2**
Walker Run	582 ± 30	530 ± 18	**661 ± 38**
Avg. DMC Suite	514 ± 65	367 ± 85	**749 ± 17**
HumanoidBench			
H1 Reach	**4129 ± 625**	2307 ± 62	4063 ± 2589
H1 Sit Hard	552 ± 71	181 ± 187	**621 ± 130**
H1 Crawl	782 ± 24	549 ± 12	**897 ± 14**
H1 Balance Simple	88 ± 29	113 ± 4	**184 ± 30**
Avg. H-Bench	1388 ± 187	788 ± 66	**1441 ± 691**

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 main contributions of GENIEO, including the discrete-time invertible dual-variable policy, tractable dummy-action likelihood/entropy computation, and empirical evaluation on DMControl and HumanoidBench. These claims are supported by the method, theoretical analysis, and experimental results reported in the paper.

   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: The conclusion includes a dedicated **Limitation and Future Work** paragraph discussing the increased dimensionality introduced by the dual-variable architecture, potential scaling challenges in very large action spaces, and sensitivity to the entropy coefficient and coherence penalty.

   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 paper states the assumptions for the change-of-variables lemma, the exact invertibility theorem, and the entropy-gradient theorem. Complete proofs are provided in Appendix A.1, Appendix A.2, and Appendix A.3.

   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: The paper describes the evaluated environments, baselines, training protocol, random seeds, and implementation details. Additional reproducibility information, including hyperparameters and baseline implementation sources, is provided in Appendix B.2.

   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: The camera-ready code archive contains the core GENIEO implementation and a README describing the main algorithmic components and dependencies. Appendix B.2 reports the experimental framework, hyperparameters, and baseline sources.

   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: The paper specifies the benchmark tasks, evaluated baselines, optimizer settings, batch size, replay buffer size, learning rates, generative steps, entropy coefficient, and other key hyperparameters in the main text and Appendix B.2.

   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: The main results are reported over 5 random seeds. Figures show mean returns with shaded bands corresponding to ± 1 standard deviation, and tables report mean $\pm$ standard deviation.

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 B.2 reports the hardware used for experiments, including 4 NVIDIA RTX 4090 GPUs, and provides an example runtime of approximately 10 hours for 1M training iterations on the Dog Run task.

   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: The research uses standard simulated continuous-control benchmarks and does not involve human subjects, private data, surveillance, or sensitive attributes. We have reviewed the NeurIPS Code of Ethics and believe the work 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: [N/A]

    Justification: This work focuses on RL algorithm design and simulated continuous control, which do not have a direct path to immediate negative societal impacts.

    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, scraped datasets, pretrained foundation models, image generators, or other assets with a high risk of misuse. The released code concerns simulated reinforcement learning experiments.

    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: The paper cites the benchmark suites and baseline repositories used in the experiments and lists their source URLs in Appendix B.2. We use these assets according to their respective open-source licenses, and the supplementary materials provide the corresponding license information when available.

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 camera-ready code archive includes the core implementation of GENIEO and an accompanying README documenting the main algorithmic components, dependencies, and usage. Appendix B.2 describes the experimental setup and hyperparameters.

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: The paper does not involve crowdsourcing, human-subject experiments, user studies, or human data collection.

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: The research does not involve human subjects, crowdsourcing, or personal data collection, so IRB approval or 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, theory, experiments, and scientific contributions do not rely on LLMs as an important, original, or non-standard component. Any language editing assistance, if used, did not affect the methodology, experiments, or scientific claims.

    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.