MALA-GUIDED MIRROR DESCENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Policy mirror descent (PMD) defines policy improvement as a simple distribution-space update: exponentially tilt the current policy toward high-value actions while remaining close to the original policy. For expressive generative policies, however, realizing this prescribed target distribution can itself be difficult. We introduce MALA-Guided Mirror Descent (MGMD), which treats policy improvement as an inference problem rather than solely as an actor-training objective. MGMD represents the diffusion policy through an energy function and uses annealed Metropolis-adjusted Langevin dynamics to traverse a sequence of value-tilted noisy distributions whose zero-noise limit is the PMD target. The resulting improved actions are used for environment interaction and Bellman targets, and are continuously distilled back into the diffusion policy with the standard denoising objective. On a controlled multimodal bandit where the exact PMD target is known, MGMD tracks the target more accurately than advantage-weighted regression as the target moves farther from the base policy. Across six MuJoCo continuous-control tasks, MGMD demonstrates consistently strong performance. These results suggest that inference-time computation can provide a persistent mechanism for policy improvement when it is directed toward a prescribed policy-improvement distribution and subsequently amortized into the policy.

1 INTRODUCTION

A central challenge in actor–critic reinforcement learning is translating value information into an improved policy (Kakade & Langford, 2002; Schulman et al., 2015). Even when the critic provides a reliable ranking of actions, learning may fail if the policy update does not accurately realize the improvement implied by the value function (Marino et al., 2021).

In the setting of diffusion policies, the problem of policy improvement in any capacity is not straightforward. Standard policy gradient methods fall short as they involve costly and noisy backpropagation through the inference procedure (Psenka et al., 2024; Ma et al., 2025). Many diffusion policy methods must circumvent this by providing signal to the policy through other techniques (Yang et al., 2023; Psenka et al., 2024; Ma et al., 2025; Ding et al., 2024).

Policy improvement can be decomposed into identifying high-value regions that receive increased mass under the target update and amortizing that improved distribution into the policy. The former can require substantial search at inference time, whereas the latter can be treated as a supervised distribution-fitting problem (Marino et al., 2021).

Related to finding high-Q regions, much work has gone into methods for producing actions from a higher-value distribution than that of the base policy by scaling compute at inference time (Zhang et al., 2026; Chalumeau et al., 2025). While these methods produce improved samples, they are generally not designed to sample from a prescribed target distribution with desirable properties as a policy-improvement operator (Zhang et al., 2026; Chalumeau et al., 2025). As a result, the compute expended on sampling improved actions does not by itself improve the underlying policy, since distilling those actions into the base policy need not correspond to a well-defined policy-improvement update.

In contrast to this, we introduce MALA-Guided Mirror Descent (MGMD), whose only source of policy improvement comes from the continuous distillation of higher-value actions produced by scaling compute at inference time. In order to achieve this, we construct a guided sampler which accurately targets the mirror descent policy update, effectively balancing optimization of actions

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

under the critic while remaining close in distribution to the base policy (Lan, 2023; Zhan et al., 2023; Tomar et al., 2022).

As a result, when the improved distribution places substantial mass in regions that are difficult to reach with the base sampler, additional inference-time computation can improve realization of the policy update, allowing for a more accurate policy improvement step during distillation.

MGMD approximately realizes the KL-regularized mirror-descent target through inference-time sampling. MGMD targets the mirror-descent distribution as the result of an energy-based annealed Markov Chain Monte Carlo (MCMC) sampler. It constructs a sequence of value-tilted diffusion distributions whose final state recovers the desired policy-improvement update, and employs Metropolis-adjusted Langevin dynamics to traverse this path from Gaussian noise to the clean-action target. In contrast to unadjusted critic guidance, the Langevin proposal is paired with a Metropolis correction that accounts for finite-step discretization error at each intermediate target (Roberts & Tweedie, 1996; Du et al., 2023). The resulting samples are used for rollouts, policy distillation and constructing temporal-difference (TD) targets, unifying inference-time policy improvement with subsequent actor and critic learning.

We evaluate MGMD by separating the quality of the policy-improvement update from its downstream effect on reinforcement learning. First, we isolate policy-improvement error in a controlled two-dimensional multimodal bandit where both the value function and the exact Mirror Descent target are known. In this context, MGMD more accurately recovers the target distribution than advantage-weighted regression, with the difference becoming more pronounced as the target distribution strays farther from that of the base policy. Second, we evaluate whether reducing this discrepancy improves online reinforcement learning performance across six MuJoCo continuous-control environments, where MGMD consistently matches or exceeds the performance of other methods despite substantial variation in baseline effectiveness. Finally, targeted ablations demonstrate that these results are not primarily due to increased sampling computation, that adaptive MALA step sizes reduce dependence on environment-specific tuning, and that the policy used to generate TD target actions can significantly influence learning.

Our contributions are threefold. First, we formulate diffusion-policy improvement as an explicit sampling problem and use an energy-parameterized policy with annealed MALA to approximately realize the prescribed PMD update. Second, we continuously amortize the resulting improved actions back into the actor while reusing them for environment interaction and Bellman targets. Third, we directly measure policy-improvement error against a known target in a controlled multimodal bandit, complementing downstream evaluation across six MuJoCo tasks.

2 Related Work

2.1 Online Reinforcement Learning with Generative Policies

A central challenge in online reinforcement learning with diffusion policies is converting critic information into an improved policy distribution. DIPO (Yang et al., 2023) performs gradient ascent on sampled actions and fits the diffusion policy to the improved actions, while Q-score matching (Psenka et al., 2024) derives an actor objective from the relationship between the policy score and the action gradient of the critic. QVPO (Ding et al., 2024) instead uses a Q-weighted variational objective together with entropy regularization and critic-based action selection. These methods are applicable to the same online control setting as MGMD, but they optimize actor objectives that do not prescribe the policy-mirror-descent distribution. QSM and QVPO additionally modify the actor objective or behavior policy, so they do not isolate accuracy in realizing a fixed mirror-descent update.

DPMD (Ma et al., 2025) is the closest objective-matched alternative. It targets the same update, $\pi_{k+1}(a \mid s) \propto \pi_k(a \mid s) \exp(\eta Q_k(s,a))$, using reweighted score matching. Thus DPMD avoids sampling from the improved distribution and instead changes the weighting of samples used to train the next policy. FPMD (Chen et al., 2025) applies the same principle to flow policies, while SiMPO (Ma et al., 2026) generalizes reweighted measure matching beyond exponential weights. MGMD differs in treating the prescribed PMD distribution as an explicit inference target before amortizing the resulting samples. DPMD shares both our policy objective and online diffusion-

policy setting and is therefore included as our primary objective-matched baseline. FPMD changes the generative policy class and inference procedure, so it does not provide the same controlled diffusion-policy comparison.

Other online diffusion actor-critic methods pursue different policy objectives. DACER (Wang et al., 2024) regulates diffusion-policy exploration through an entropy estimator, while DIME (Celik et al., 2025) and MaxEntDP (Dong et al., 2025) optimize maximum-entropy policies. These methods address the same control environments but do not isolate realization of the PMD operator, and their published algorithms also introduce distinct entropy or critic machinery.

2.2 Guided Policy Improvement

Guidance provides an alternative way to incorporate value information during generative-policy inference. Training-Free Guidance (Ye et al., 2024) studies conditional diffusion sampling from a fixed model using an external predictor without additional guidance training. QGPO (Lu et al., 2023) specializes energy guidance to offline reinforcement learning and derives the intermediate guidance associated with an exponentially tilted behavior policy. Its population guarantee requires learning an additional time-dependent energy model, and its experiments assume an offline dataset and fixed behavior distribution. It therefore does not address online learning from scratch with a continually changing reference policy and critic.

CGPO (Ding et al., 2026) applies training-free critic guidance directly during online diffusion denoising and regresses the actor toward the guided actions, although it additionally couples guidance with a distributional quantile critic, DDQN-style targets, and additional value-calibration machinery. Our controlled experiments instead use a shared twin-Q critic to isolate differences in policy improvement.

QPILOTS (Ruan et al., 2026) similarly uses critic gradients to steer a generative policy at inference time, with an idealized construction based on a KL-regularized exponential tilt. Its main setting performs test-time steering of an existing flow policy rather than repeated online distillation of PMD updates, so it is not directly comparable to our online-from-scratch setting.

2.3 MCMC and Diffusion Guidance

Our sampler builds on two lines of work in guided diffusion. Training-free guidance methods use gradients of external objectives to alter the reverse process without retraining the base model (Ye et al., 2024; Bansal et al., 2023). Separately, Du et al. (2023) introduce energy-parameterized diffusion models and show that MCMC, including Metropolis-corrected samplers, can improve compositional and guided generation. Their setting concerns sampling from fixed generative models rather than policy improvement in reinforcement learning. MGMD combines these ideas with policy mirror descent by using the evolving policy energy and critic to define the MCMC target inside an online actor-critic algorithm.

3 Preliminaries

We consider an infinite-horizon discounted Markov decision process (MDP) defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \rho_0, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, P is the transition kernel, r is the reward function, ρ_0 is the initial-state distribution, and $\gamma \in (0, 1)$ is the discount factor. A policy $\pi(a \mid s)$ specifies a distribution over actions conditioned on the current state. Starting from $s_0 \sim \rho_0$, the agent samples $a_t \sim \pi(\cdot \mid s_t)$, receives reward $r(s_t, a_t)$, and transitions according to $s_{t+1} \sim P(\cdot \mid s_t, a_t)$. The value function is $V^\pi(s) = \mathbb{E}_{\pi,P}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s]$, and the action-value function is $Q^\pi(s, a) = \mathbb{E}_{\pi,P}[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, a_0 = a]$. The objective is to find a policy that maximizes the expected discounted return, $J(\pi) = \mathbb{E}_{s_0 \sim \rho_0}[V^\pi(s_0)]$. This casts policy learning as an optimization problem over the space of policies, motivating iterative policy-improvement methods such as policy mirror descent.

Policy Mirror Descent. Mirror descent is a first-order optimization (Beck & Teboulle, 2003) method that updates the current iterate by optimizing a local linear approximation of the objective while penalizing large changes from the previous iterate. The geometry of the update is determined

by a Bregman divergence, allowing the optimization step to respect the structure of the underlying domain (Bregman, 1967; Beck & Teboulle, 2003). For policies, which lie on the probability simplex, choosing negative entropy as the mirror map yields a KL-divergence proximity term and the update

$$\pi_{k+1}(\cdot \mid s) = \arg\max_{\pi(\cdot|s)} \left\{ \mathbb{E}_{a\sim\pi(\cdot|s)}[Q^{\pi_k}(s,a)] - \frac{1}{\eta} D_{\mathrm{KL}}(\pi(\cdot \mid s) \,\|\, \pi_k(\cdot \mid s)) \right\}.$$

The first term favors distributions that place more probability mass on high-value actions, while the KL term limits how far the update moves from the current policy. This tradeoff gives a multiplicative update in distribution space rather than an additive update in parameter space. The KL-regularized optimization can be solved exactly in policy space, yielding an exponential reweighting of the current policy (Lan, 2023; Zhan et al., 2023):

$$\pi_{k+1}(a \mid s) = \frac{\pi_k(a \mid s)\exp(\eta Q^{\pi_k}(s,a))}{\int_{\mathcal{A}} \pi_k(a' \mid s)\exp(\eta Q^{\pi_k}(s,a'))\, da'}.$$

This distribution-space update is attractive not only for its simple closed form but also for its theoretical guarantees. In the tabular setting, natural policy gradient can be viewed as a mirror-descent update in policy space (Xiao, 2022), and Agarwal et al. (2021) establish an $\mathcal{O}(1/K)$ convergence rate to the global optimum after K iterations. These results motivate policy mirror descent as a principled framework for policy optimization while also highlighting a practical gap. The theory characterizes the idealized target distribution but does not specify how to realize or sample from that target when the policy is a complex parameterized generative model. For diffusion policies in particular, representing and sampling from the exponentially tilted target is nontrivial.

Diffusion Models Diffusion models (Sohl-Dickstein et al., 2015; Ho et al., 2020) represent complex continuous distributions by gradually corrupting data with Gaussian noise and learning to reverse this process. In our setting, let x_0 denote a clean action drawn from the policy at state s. Given a noise schedule $\{\alpha_t\}_{t=1}^T$ with $\bar{\alpha}_t = \prod_{i=1}^t \alpha_i$, the forward process satisfies $x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$, where $\epsilon \sim \mathcal{N}(0, I)$ (Ho et al., 2020). As the noise level increases, the distribution approaches a standard Gaussian, from which the reverse process iteratively denoises to produce an action.

The reverse process is parameterized by a noise predictor $\hat{\epsilon}_\theta(x_t, s, t)$ that estimates the Gaussian noise added to the clean action and is trained using the standard denoising objective (Ho et al., 2020). At its population optimum, the noise predictor is proportional to the score of the noisy action distribution, so denoising can equivalently be viewed as learning the gradient of its log density (Ho et al., 2020; Song et al., 2021). Given a noisy action and the predicted noise, the forward noising equation can also be inverted to estimate the corresponding clean action using Tweedie's formula (Efron, 2011; Ho et al., 2020).

Metropolis-Adjusted Langevin Algorithm To sample from an unnormalized density $\rho(x) \propto \exp(-E(x))$, the Metropolis-Adjusted Langevin Algorithm (MALA) uses gradient-based Langevin proposals together with a Metropolis–Hastings correction. The correction removes the finite-step discretization bias of the Langevin proposal while requiring only ratios of the target density, so the normalizing constant of ρ need not be known.

Applying this correction to diffusion policies requires evaluating density ratios of the noisy policy marginal. This is problematic because standard diffusion models parameterize only the score or noise network $\hat{\epsilon}_\theta(x_t, s, t)$, providing access to gradients but not density values. To resolve this, we express the noisy policy marginal in energy-based form as $p_{\theta,t}(x_t \mid s) \propto \exp(-E_\theta(s, x_t, t))$, in which case the predicted noise corresponds to the gradient of the energy network, $\hat{\epsilon}_\theta(s, x_t, t) = \sigma_t \nabla_{x_t} E_\theta(s, x_t, t)$, where $\sigma_t = \sqrt{1-\bar{\alpha}_t}$. This representation preserves the standard denoising objective while providing the unnormalized density ratios $p_{\theta,t}(y \mid s)/p_{\theta,t}(x \mid s) = \exp(E_\theta(s,x,t) - E_\theta(s,y,t))$ required by the Metropolis correction.

4 MGMD

4.1 Annealed MALA Sampling

The mirror-descent target is defined on clean actions, whereas diffusion sampling evolves through noisy intermediate states. We therefore construct a sequence of tilted distributions over diffusion

Algorithm 1: TILTEDSAMPLE: annealed MALA sampling of the mirror-descent target

Input: state s; energy policy E_θ; critics $\bar{Q}_\phi$; step size η; log-step-scales $\{\ell_t\} \in \mathbb{R}^T$
Data: levels T, schedule $\{\bar{\alpha}_t, \alpha_t\}$, MALA steps K, target acceptance $a^\star$, adaptation rate κ
$x \sim \mathcal{N}(0, I)$
for $t \leftarrow T-1$ **to** 0 **do**
  $h_t \leftarrow e^{\ell_t}(1-\alpha_t)$
  for $k \leftarrow 1$ **to** K **do**
    $E \leftarrow E_t^{\text{guided}}(x)$
    $x' \leftarrow x - h_t \nabla_x E_t^{\text{guided}}(x) + \sqrt{2h_t}\xi$
    $E' \leftarrow E_t^{\text{guided}}(x')$
    $\log A \leftarrow (E - E') + \dfrac{\|x' - x + h_t \nabla_x E_t^{\text{guided}}(x)\|^2 - \|x - x' + h_t \nabla_{x'} E_t^{\text{guided}}(x')\|^2}{4h_t}$
    $x \leftarrow x'$ with probability $\min\{1, e^{\log A}\}$, else $x \leftarrow x$
    $\ell_t \leftarrow \ell_t + \kappa(\hat{a}_t - a^\star)$
$\hat{\epsilon}^{\text{guided}} \leftarrow \hat{\epsilon}_\theta(s, x, 0) - \eta_0 \, \sigma_0 \, \nabla_x \widetilde{Q}\big(s, \hat{x}_0(s, x, 0)\big)$
return $a = (x - \sigma_0 \hat{\epsilon}^{\text{guided}})/\sqrt{\bar{\alpha}_0}$

levels whose clean endpoint recovers the desired mirror-descent update. Algorithm 1 traverses this sequence from Gaussian noise to the clean-action target using annealed MALA. MGMD uses the energy-parameterized diffusion policy introduced in the preliminaries, which provides both the score used in Langevin proposals and the unnormalized density ratios required by the Metropolis correction. Combining this parameterization with the Tweedie estimate gives

$$\hat{x}_0(s, x, t) = \frac{x - \sigma_t \hat{\epsilon}_\theta(s, x, t)}{\sqrt{\bar{\alpha}_t}} = \frac{x - \sigma_t^2 \nabla_x E_\theta(s, x, t)}{\sqrt{\bar{\alpha}_t}}.$$

The factor $\eta_t = \eta \bar{\alpha}_t$ gradually turns on value guidance as the diffusion state approaches the clean-action space. We use this simple schedule so that guidance vanishes in the high-noise regime, where the clean-action estimate is least informative, and reaches the desired mirror-descent strength at the clean endpoint. Other annealing schedules are possible; we use this choice throughout our experiments.

Guidance signal. We maintain twin critics Q_{ϕ_1} and Q_{ϕ_2} and use their average $\bar{Q}_\phi = \frac{1}{2}(Q_{\phi_1} + Q_{\phi_2})$ inside the sampler. Because critic scales can vary substantially across environments, we normalize this aggregate before applying the mirror-descent tilt. Let $\hat{\mu}$ and $\hat{\sigma}$ denote slow exponential moving averages of its batch mean and standard deviation, and define $\widetilde{Q}(s, a) = (\bar{Q}_\phi(s, a) - \hat{\mu})/\hat{\sigma}$. The shift $\hat{\mu}$ cancels from both gradients and energy differences, while scaling by $\hat{\sigma}$ makes the mirror-descent step size η less sensitive to the scale of the critic and allows a common value to transfer across environments.

The mirror-descent target is defined on clean actions, while diffusion sampling proceeds through noisy intermediate states. We therefore extend the clean-action tilt to each diffusion level by evaluating the normalized critic on the Tweedie estimate,

$$\rho_t(x \mid s) \propto p_{\theta,t}(x \mid s) \exp\Big(\eta_t \, \widetilde{Q}\big(s, \hat{x}_0(s, x, t)\big)\Big), \qquad \eta_t = \eta \bar{\alpha}_t. \tag{1}$$

For fixed h_t, each MALA transition at level t leaves ρ_t invariant. After the K updates at level t, the resulting state initializes the chain at level $t-1$.

This target combines the noisy policy marginal with a value-dependent tilt whose strength varies across diffusion levels. Since the policy is represented through its energy, the same target can be expressed through its energy function

$$E_t^{\text{guided}}(x) = E_\theta(s, x, t) - \eta_t \, \widetilde{Q}\big(s, \hat{x}_0(s, x, t)\big).$$

Although the intermediate distributions in equation 1 are a designed annealing path rather than the exact noisy marginals of the clean tilted policy, the construction matches the desired distributions at

both endpoints. At high noise, $\bar{\alpha}_t \to 0$, so $\eta_t \to 0$ and $\rho_t \to p_{\theta,t} \to \mathcal{N}(0, I)$, allowing the chain to be initialized from Gaussian noise. At the clean endpoint, $\bar{\alpha}_0 \to 1$ and $\hat{x}_0 \to x$, giving

$$\rho_0(x \mid s) \propto \pi_\theta(x \mid s) \exp\big(\eta\, \widetilde{Q}(s, x)\big).$$

which recovers the desired mirror-descent tilt. We therefore approximate sampling from the clean PMD target by traversing this designed sequence of intermediate distributions from Gaussian noise to the desired clean-action tilt. MALA provides a natural mechanism for this traversal by refining the current sample toward each successive target while correcting the finite-step Langevin bias.

MALA sampling across noise levels. At each diffusion level, we run K MALA steps targeting ρ_t. Starting from the current state x, we construct the Langevin proposal and its corresponding Gaussian proposal distribution,

$$x' = x - h_t \nabla_x E_t^{\text{guided}}(x) + \sqrt{2h_t}\, \xi, \qquad \xi \sim \mathcal{N}(0, I),$$
$$q_t(x' \mid x) = \mathcal{N}\big(\mu_t(x), 2h_t I\big), \qquad \mu_t(x) = x - h_t \nabla_x E_t^{\text{guided}}(x).$$

A finite-step Langevin proposal does not in general preserve the desired target exactly, so MALA applies the Metropolis–Hastings correction

$$A(x, x') = \min\left\{1, \exp\left(E_t^{\text{guided}}(x) - E_t^{\text{guided}}(x') + \frac{\|x' - \mu_t(x)\|^2 - \|x - \mu_t(x')\|^2}{4h_t}\right)\right\}.$$

Only ratios of ρ_t appear in this acceptance probability, so the intractable normalizing constant of equation 1 need not be evaluated. After the K MALA updates, the accepted state initializes the chain at level $t-1$, relying on adjacent targets being sufficiently close that a sample from ρ_t provides a useful initialization for ρ_{t-1}.

Adaptive step sizes. The efficiency of MALA depends on the step size and acceptance rate, whose appropriate scale varies across diffusion levels, environments, and training iterations. We parameterize $h_t = e^{\ell_t}(1 - \alpha_t)$ and update the log-scale after each MALA step using $\ell_t \leftarrow \ell_t + \kappa(\hat{a}_t - a^\star)$, where $\hat{a}_t$ is the empirical acceptance fraction across the current minibatch at noise level t. This adaptation targets acceptance rate $a^\star$ while allowing the sampler to track changes in the policy and critic during training.

4.2 Learning from Tilted Actions

At iteration k, the current critic defines the mirror-descent improvement $\tilde{\pi}_k(a \mid s) \propto \pi_k(a \mid s) \exp(\eta\, \widetilde{Q}_k(s, a))$. After each environment step, we sample a minibatch of transitions $(s, a, r, s', d) \sim \mathcal{D}$ and draw one tilted action for each next state, $a' \sim \tilde{\pi}(\cdot \mid s')$, where $\tilde{\pi}$ denotes the sampler in Algorithm 1. The same tilted actions are reused for both critic learning and policy distillation in Algorithm 2.

We fit the diffusion policy to the tilted actions using the standard denoising objective,

$$\mathcal{L}_\pi(\theta) = \mathbb{E}\Big[\big\|\hat{\epsilon}_\theta\big(s', \sqrt{\bar{\alpha}_t} a' + \sigma_t \epsilon, t\big) - \epsilon\big\|_2^2\Big]. \tag{2}$$

Here $t \sim \mathcal{U}\{0, \ldots, T-1\}$ and $\epsilon \sim \mathcal{N}(0, I)$. Because the actions are already sampled from the tilted distribution, the policy can be trained with the ordinary unweighted denoising loss without the need for importance weighting.

For the critic, we construct Bellman targets reusing the same tilted actions,

$$y = r + \gamma(1 - d) \min_{i \in \{1,2\}} Q_{\bar{\phi}_i}(s', a'), \qquad \mathcal{L}_Q(\phi_i) = \mathbb{E}\big[(Q_{\phi_i}(s, a) - y)^2\big].$$

At first, this may seem like an unusual choice, but it is more than just a cost-saving measure. If we re-frame the algorithm to consider the policy to be the distribution of tilted actions rather than the distribution parameterized by the policy network, then the training scheme is much more analogous to standard reinforcement learning algorithms. Under this view, rollouts using tilted actions are rollouts of the current policy, training the critic on tilted actions becomes policy evaluation, and the distillation of tilted actions becomes the policy improvement step. Another view is that the algorithm distills and learns the critic for the tilted policy simultaneously rather than having to learn Q based on samples from the current policy, which would introduce lag between the policy and the critic's evaluation.

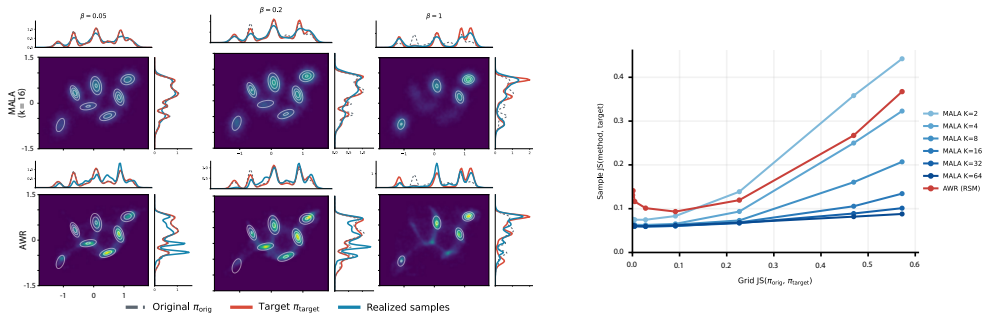


Figure 1: **Realizing the mirror-descent target on a 2D multimodal bandit with a known advantage function**. Left, samples from MGMD (top, K=16) and advantage-weighted regression (AWR, bottom) at increasing tilt strength, β, overlaid on the exact mirror descent target distribution. AWR, the policy update used by DPMD with the critic replaced by the advantage, fits the diffusion policy by reweighting the denoising loss by $\exp(\beta A)$. Right, Jensen-Shannon divergence between each method's samples and the target, as a function of the distance from the policy to the target. MGMD's error is roughly independent of the divergence between the target and base policy, whereas AWR degrades with the divergence.

5 Experiments

We evaluate MGMD along two dimensions. First, we ask whether more accurately realizing the mirror-descent target translates into improved online reinforcement-learning performance across continuous-control tasks. Second, we use targeted ablations to study which components of the method are responsible for these gains, including adaptive MALA step sizes, constructing critic targets from tilted actions, and the additional inference-time computation used by the sampler.

5.1 Experimental Setup

2-Dimensional Bandit Task To isolate error in realizing the PMD update from critic estimation, we construct a multimodal continuous bandit on $x \in \mathbb{R}^2$ with a known advantage function, allowing the target $p^*(x) \propto p_0(x)\exp(\beta A(x))$ to be computed exactly. We compare MGMD against advantage-weighted regression (AWR), which fits a diffusion model using a denoising loss weighted by $\exp(\beta A(x))$, and measure Jensen–Shannon divergence from p^*. Complete task and training details are given in Appendix B.1.

MuJoCo Continuous Control Environments We evaluate MGMD on six MuJoCo-v3 continuous-control environments and compare against SAC (Haarnoja et al., 2018), DIPO (Yang et al., 2023), and DPMD (Ma et al., 2025). Our goal is a controlled comparison of policy-improvement mechanisms rather than an exhaustive benchmark over actor-critic architectures. SAC provides a standard Gaussian actor–critic baseline, while DIPO and DPMD use diffusion policies and incorporate the critic through action optimization and mirror-descent-style reweighting, respectively. MGMD, DIPO, and DPMD therefore share the diffusion-policy setting but differ primarily in how policy improvement is realized. We hold the interaction budget and update frequency fixed across methods, report results over 10 independent random seeds, and evaluate every 10^4 environment steps using 10 rollout episodes. All methods are trained for 10^6 environment interactions. For diffusion-based methods, evaluation uses each algorithm's own action-selection procedure.

We align the critic and policy architectures across diffusion methods where possible, with complete architecture and hyperparameter details given in Appendix B.2. MGMD adds a final scalar-output layer to parameterize the policy energy. Although this parameterization constrains the learned score field to be the gradient of a potential, prior comparisons found such models to perform comparably to or worse than unconstrained score parameterizations in unconditional generation, suggesting that the performance of MGMD is not the result of this design choice alone (Salimans & Ho, 2021; Du et al., 2023). The remaining differences follow the published defaults.

5.2 Main Results

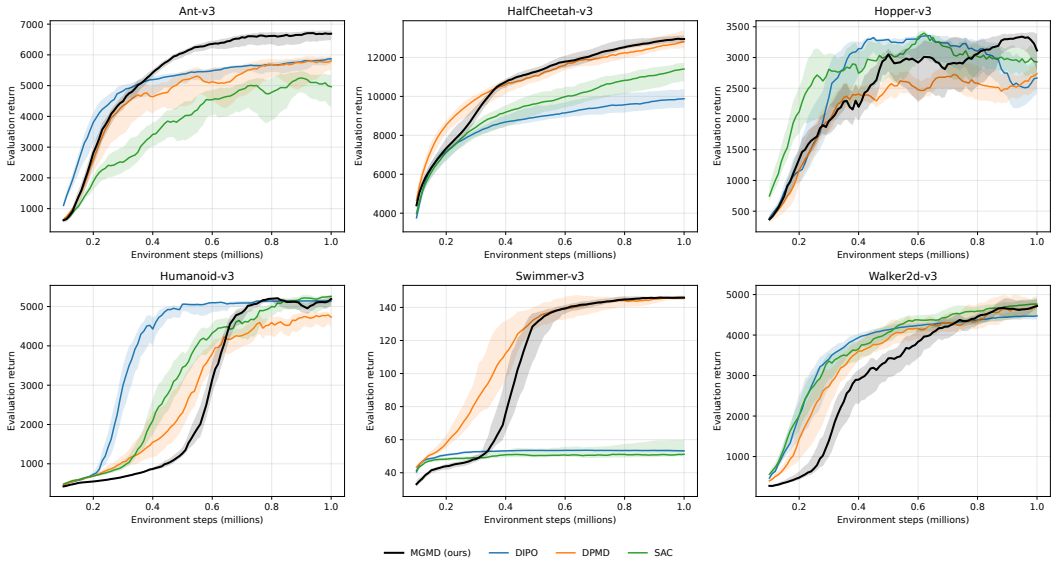


Figure 2: Median performance over 10 seeds for 6 environments. For each algorithm, we report the average over the last 10 evaluation checkpoints of the median performance over 10 seeds and an 80% confidence interval.

Environment	MGMD (ours)	DIPO	DPMD	SAC
Ant-v3	**6688** [6486, 6786]	5872 [5702, 5921]	5805 [5504, 5972]	4966 [4306, 5373]
HalfCheetah-v3	**12949** [12768, 13126]	9878 [9415, 10367]	**12808** [12487, 13394]	11407 [10777, 11718]
Hopper-v3	**3115** [2780, 3381]	2666 [2417, 2953]	2735 [2483, 2910]	**2929** [2694, 3134]
Humanoid-v3	**5194** [5033, 5262]	**5144** [5095, 5158]	4731 [4471, 4810]	**5260** [5001, 5294]
Swimmer-v3	**145.9** [144.8, 147.0]	53.2 [52.5, 53.8]	**146.0** [144.8, 147.2]	51.2 [50.1, 59.7]
Walker2d-v3	**4719** [4639, 4891]	4474 [4387, 4630]	**4705** [4473, 4969]	**4764** [4475, 4907]

Table 1: Median of evaluation return over 10 seeds, averaged over final 10 evaluation checkpoints with 80% confidence interval from bootstrapping over seeds. Figures are bold if they lie within the confidence interval of the best performing method for that environment.

Didactic Experiment Figure 1 compares MGMD with AWR on the controlled 2D bandit task described above. At weak tilt, $\beta \approx 0.05$, both methods recover the true distribution: equally capturing the modes and relative probability mass, however, as β increases AWR loses low probability modes and incorrectly estimates the probability mass of those modes that it does capture. In contrast, MGMD continues tracking the true modes and their relative mass up to the highest tilt we tested. Figure 1 quantifies this via the JS divergence between the target and realized distributions against the same divergence between the base and target distribution.

Two trends are visible in this plot. For $K \geq 16$, MGMD's error is nearly flat, barely rising from ~ 0.06 at $\beta = 0$ to ~ 0.1 at $\beta = 10$. At moderate MALA steps per level ($K \leq 8$), MGMD attains the same floor, but only at weak tilting levels. AWR, behaves differently at both ends of tilt spectrum. AWR exhibts an asymmetric U-shaped curve. Importantly at high tilts the gap between MGMD and AWR is substantial.

The controlled bandit isolates error in realizing the mirror-descent target by removing critic estimation and sequential decision making, whereas the MuJoCo experiments additionally depend on critic accuracy, exploration, bootstrapping, and optimization dynamics. We therefore interpret the bandit

as evidence that explicit sampling can reduce policy-improvement error, while the MuJoCo results test whether the resulting algorithm remains effective when these sources of error interact.

MuJoCo Across the six MuJoCo environments, MGMD remains at or near the strongest-performing method on every task. It attains the highest median return on Ant, HalfCheetah, and Hopper, and lies within the confidence interval of the best-performing method on Humanoid, Swimmer, and Walker2d. As shown in Figure 2 and Table 1, the identity of the strongest competing method varies substantially across environments, whereas MGMD maintains consistently strong performance across the full suite. This suggests that MGMD provides robust performance across tasks rather than deriving its overall performance from particularly large gains on only a subset of environments.

5.3 Ablations

We next isolate three practical design choices underlying MGMD. We test whether its gains persist under matched sampling compute and whether adaptive MALA step sizes reduce dependence on manual tuning.

Are MGMD's gains explained by additional sampling compute? We rerun MGMD with 50 diffusion steps instead of 100, reducing the sampling cost while more closely matching the baselines in terms of score evaluations. As shown in Figure 3, performance is largely preserved across all six environments. The 50-step variant closely tracks the 100-step version on most tasks, with the largest differences appearing in Hopper and in the learning speed on Humanoid. This suggests that MGMD's performance is not primarily explained by the additional sampling compute associated with using more diffusion levels.

Does adaptive step-size selection reduce tuning dependence? Following prior work that applies MALA with fixed step-size schedules, we replace the adaptive update with $h_t = c(1 - \alpha_t)$ for several fixed values of c. As shown in Figure 4, some choices of c match or even outperform the adaptive variant on individual environments, but no single value performs consistently well across all four tasks. The preferred scale varies substantially by environment, suggesting that the adaptive update primarily serves to reduce sensitivity to manual step-size tuning while maintaining strong performance across tasks.

Should the critic evaluate the improved policy? MGMD explicitly samples from the mirror-descent-improved policy, which allows the same tilted actions to be used when constructing Bellman targets. We ablate this choice by instead sampling next actions from the base diffusion policy. As shown in Figure 5, this modification lowers performance on Ant and HalfCheetah and leads to a particularly large degradation on Humanoid, while having little effect on Walker2d. These results suggest that evaluating the improved policy can reduce the lag between policy improvement and policy evaluation, which becomes especially important when the tilted policy differs substantially from the current base policy.

6 Conclusion

We introduced MALA-Guided Mirror Descent, which treats policy improvement for diffusion policies as an explicit inference problem. MGMD uses annealed Metropolis-adjusted Langevin dynamics to approximately traverse a sequence of value-tilted distributions whose zero-noise limit is the mirror-descent target, and continuously distills the resulting improved actions back into the policy. In a controlled setting, MGMD more accurately realizes the desired update as the target moves farther from the base policy, while experiments across six MuJoCo tasks demonstrate consistently strong online reinforcement-learning performance. This comes at the cost of additional inference-time computation and an energy-parameterized policy; moreover, the intermediate targets are a designed annealing path and finite MALA mixing remains a source of approximation error despite the Metropolis correction. Our evaluation is currently limited to continuous-control MuJoCo tasks, but the results suggest that inference-time computation can serve as a persistent policy-improvement mechanism when its outputs are subsequently amortized into the policy.

AI USE STATEMENT

In this work, we used generative AI tools to provide feedback on research methodology and experiments, but not to design the research methodology or experiments. We also used generative AI tools for implementation of methods, assistance with translation, and reformatting datasets.

We have not used generative AI tools for help in developing theoretical models or conceptual frameworks, formulating mathematical claims, proposing or refining hypotheses, or interpretation of results.

Generation of synthetic datasets, provision of critical ingredients for proving mathematical claims, assistance in the writing of proofs, cleaning datasets, and support in qualitative and thematic data analysis are not applicable to this work.

Additionally, generative AI tools were used to surface potentially related work and polish final wording.

We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by two authors. The paper's final text was checked by multiple authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021.

Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 843–852, 2023.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003. doi: 10.1016/S0167-6377(02)00231-6.

L. M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967. doi: 10.1016/0041-5553(67)90040-7.

Onur Celik, Zechu Li, Denis Blessing, Ge Li, Daniel Palenicek, Jan Peters, Georgia Chalvatzaki, and Gerhard Neumann. DIME: Diffusion-based maximum entropy reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 6958–6977. PMLR, 2025.

Félix Chalumeau, Daniel Rajaonarivonivelomanantsoa, Ruan John de Kock, Juan Formanek, Sasha Abramowitz, Omayma Mahjoub, Wiem Khlifi, Simon Du Toit, Louay Nessir, Refiloe Shabe, Arnol Fokam, Siddarth Singh, Ulrich Armel Mbou Sob, and Arnu Pretorius. Breaking the performance ceiling in reinforcement learning requires inference strategies. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Tianyi Chen, Haitong Ma, Na Li, Kai Wang, and Bo Dai. One-step flow policy mirror descent. *arXiv preprint arXiv:2507.23675*, 2025.

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. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Shutong Ding, Zejia Zhong, Zhongyi Wang, Ke Hu, Bikang Pan, Jingya Wang, and Ye Shi. Sample-efficient diffusion-based reinforcement learning with critic guidance. In *International Conference on Machine Learning*, 2026.

Xiaoyi Dong, Jian Cheng, and Xi Sheryl Zhang. Maximum entropy reinforcement learning with diffusion policy. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 13963–13983. PMLR, 2025.

Yilun Du, Conor Durkan, Robin Strudel, Joshua B. Tenenbaum, Sander Dieleman, Rob Fergus, Jascha Sohl-Dickstein, Arnaud Doucet, and Will Sussman Grathwohl. Reduce, reuse, recycle: Compositional generation with energy-based diffusion models and MCMC. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 8489–8510. PMLR, 2023.

Bradley Efron. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011. doi: 10.1198/jasa.2011.tm11181.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pp. 1861–1870, 2018.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning*, pp. 267–274, 2002.

Guanghui Lan. Policy mirror descent for reinforcement learning: Linear convergence, new sampling complexity, and generalized problem classes. *Mathematical Programming*, 198(1):1059–1106, 2023.

Cheng Lu, Huayu Chen, Jianfei Chen, Hang Su, Chongxuan Li, and Jun Zhu. Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 22825–22855. PMLR, 2023.

Haitong Ma, Tianyi Chen, Kai Wang, Na Li, and Bo Dai. Efficient online reinforcement learning for diffusion policy. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 41837–41853. PMLR, 2025.

Haitong Ma, Chenxiao Gao, Tianyi Chen, Na Li, and Bo Dai. SiMPO: Measure matching for online diffusion reinforcement learning. *arXiv preprint arXiv:2603.10250*, 2026.

Joseph Marino, Alexandre Piché, Alessandro Davide Ialongo, and Yisong Yue. Iterative amortized policy optimization. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Michael Psenka, Alejandro Escontrela, Pieter Abbeel, and Yi Ma. Learning a diffusion model policy from rewards via q-score matching. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 41163–41182. PMLR, 2024.

Gareth O Roberts and Richard L Tweedie. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363, 1996.

Yifan Ruan, Chenyang Cao, Andreas Burger, Ali Pesaranghader, Kaveh Kamali, Jaehong Kim, Nandita Vijaykumar, Alan Aspuru-Guzik, Igor Gilitschenski, and Nicholas Rhinehart. QPILOTS: Efficient test-time q-steering for flow policies. *arXiv preprint arXiv:2606.14801*, 2026.

Tim Salimans and Jonathan Ho. Should EBMs model the energy or the score? In *Energy Based Models Workshop at ICLR*, 2021.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International Conference on Machine Learning*, pp. 1889–1897, 2015.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265. PMLR, 2015.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Manan Tomar, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh. Mirror descent policy optimization. In *International Conference on Learning Representations*, 2022.

Yinuo Wang, Likun Wang, Yuxuan Jiang, Wenjun Zou, Tong Liu, Xujie Song, Wenxuan Wang, Liming Xiao, Jiang Wu, Jingliang Duan, and Shengbo Eben Li. Diffusion actor-critic with entropy regulator. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Lin Xiao. On the convergence rates of policy gradient methods. *Journal of Machine Learning Research*, 23(282):1–36, 2022.

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.

Haotian Ye, Haowei Lin, Jiaqi Han, Minkai Xu, Sheng Liu, Yitao Liang, Jianzhu Ma, James Zou, and Stefano Ermon. TFG: Unified training-free guidance for diffusion models. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Wenhao Zhan, Shicong Cen, Baihe Huang, Yuxin Chen, Jason D. Lee, and Yuejie Chi. Policy mirror descent for regularized reinforcement learning: A generalized framework with linear convergence. *SIAM Journal on Optimization*, 33(2):1061–1091, 2023. doi: 10.1137/21M1456789.

XiangCheng Zhang, Haowei Lin, Haotian Ye, James Y. Zou, Jianzhu Ma, Yitao Liang, and Yilun Du. Inference-time scaling of diffusion models through classical search. In *International Conference on Learning Representations*, 2026.

A ALGORITHM

Algorithm 2: MGMD

Input: mirror-descent step size η; discount γ; batch size B; Polyak rate τ
Initialize energy policy θ, critics ϕ_1, ϕ_2, target critics $\bar{\phi}_i \leftarrow \phi_i$, log-scales $\ell \leftarrow 0$, and replay buffer $\mathcal{D}$;
for *each environment step* **do**
  Sample $\{(s_j, a_j, r_j, s'_j, d_j)\}_{j=1}^{B} \sim \mathcal{D}$;
  set $s'_B \leftarrow s_{\text{env}}$,
  $a'_j \leftarrow \textsc{TiltedSample}(s'_j, E_\theta, \{\phi_i\}, \eta, \ell)$ for $j = 1, \dots, B$;
  $y_j \leftarrow r_j + \gamma(1 - d_j) \min_i Q_{\bar{\phi}_i}(s'_j, a'_j)$;
  Update ϕ_1, ϕ_2 using $\frac{1}{B-1} \sum_{j<B} (Q_{\phi_i}(s_j, a_j) - y_j)^2$;
  $\bar{\phi}_i \leftarrow \tau\phi_i + (1 - \tau)\bar{\phi}_i$;
  Update $(\hat{\mu}, \hat{\sigma})$ from $Q_{\text{avg},\phi}(s'_j, a'_j)$;
  Update θ by diffusion regression on $\{(s'_j, a'_j)\}_{j=1}^{B}$;
  Execute a'_B and store the resulting transition in $\mathcal{D}$;

B EXPERIMENT DETAILS

B.1 2D BANDIT

The base policy is a seven-component Gaussian mixture

$$p_0(\mathbf{x}) = \sum_{i=1}^{7} w_i \, \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_i, \Sigma_i),$$

with

$$\Sigma_i = \begin{pmatrix} s_{i1}^2 & \rho_i s_{i1} s_{i2} \\ \rho_i s_{i1} s_{i2} & s_{i2}^2 \end{pmatrix}.$$

The parameters are listed in Table 2.

The reward is a sum of eight Gaussian bumps plus a sinusoidal and polynomial term,

$$R(\mathbf{x}) = \sum_{j=1}^{8} a_j \exp\left[-\frac{1}{2} \sum_{k=1}^{2} \left(\frac{x_k - c_{jk}}{h_{jk}} \right)^2 \right] + 0.10 \sin(11x_1 + 15x_2 + 0.45) - 0.025\|\mathbf{x}\|^2 - 0.003\|\mathbf{x}\|^4.$$

The parameters of the eight Gaussian bumps are listed in Table 3.

The scalar advantage is computed as $A(x) = (R(x) - \mu)/\sigma$, with μ and σ the mean and standard deviation of R under the base policy p_0. The PMD target is $p^*(x) \propto p_0(x) \exp(\beta A(x))$, where β plays the role of η, the mirror descent step size.

We train an energy-based diffusion model on $N = 5000$ samples from p_0 with 20 diffusion steps and compare MGMD, with K MALA correction steps per level, against advantage-weighted regression (AWR), where a second diffusion model is fitted to $N = 5000$ samples from the base policy using a denoising loss weighted by $\exp(\beta A(x))$. Both methods draw 20000 samples, which we compare to the target distribution $p^*(x)$ using the Jensen–Shannon divergence.

B.2 MUJOCO

All four methods use twin-Q critics trained by off-policy TD learning with discount $\gamma = 0.99$, Polyak target averaging with $\tau = 0.005$, batch size 256, replay-buffer capacity 2.5×10^5, and one gradient update per environment interaction. Training begins after 3×10^4 initial environment interactions. MGMD uses Adam with actor and critic learning rates 3×10^{-4} and 1.5×10^{-4}, respectively.

The diffusion methods use critics with three hidden layers of width 256 and diffusion-policy MLPs with three hidden layers of width 256, with Mish activations. SAC uses ReLU activations. Diffusion-policy inputs concatenate the state, noisy action, and a sinusoidal embedding of the diffusion level. All diffusion methods use $T = 100$ noise levels; DIPO uses a 32-dimensional noise embedding compared with 16 dimensions for MGMD and DPMD, while DPMD uses a linear β schedule and MGMD and DIPO use a cosine schedule. MGMD additionally uses a scalar-output energy parameterization of the policy.

For MGMD, the mirror-descent strength is $\eta = 64$, with $K = 1$ MALA update per diffusion level. The adaptive MALA step sizes target acceptance rate $a^\star = 0.7$ using Robbins–Monro rate $\kappa = 0.2$. Tweedie estimates used for critic guidance and final actions are clipped to the environment action bounds. Unless otherwise specified above, baseline-specific choices follow their published defaults.

Table 2: Parameters of the base policy p_0.

i	w_i	μ_{i1}	μ_{i2}	s_{i1}	s_{i2}	ρ_i
1	0.08	−1.10	−0.72	0.10	0.16	0.45
2	0.16	−0.62	0.34	0.07	0.11	−0.55
3	0.10	−0.18	−0.10	0.16	0.07	0.35
4	0.22	0.08	0.58	0.09	0.15	−0.25
5	0.13	0.46	−0.42	0.14	0.08	0.50
6	0.18	0.83	0.20	0.08	0.13	−0.40
7	0.13	1.12	0.78	0.12	0.09	0.20

Table 3: Parameters of the Gaussian bumps in the reward $R(\mathbf{x})$.

j	a_j	c_{j1}	c_{j2}	h_{j1}	h_{j2}
1	2.9	−1.08	−0.66	0.13	0.16
2	−2.6	−0.62	0.34	0.11	0.14
3	−2.0	−0.18	−0.10	0.18	0.09
4	1.2	0.10	0.58	0.12	0.17
5	−2.7	0.46	−0.42	0.15	0.10
6	1.9	0.83	0.20	0.12	0.15
7	3.2	1.12	0.78	0.16	0.13
8	−1.2	0.34	0.02	0.22	0.18

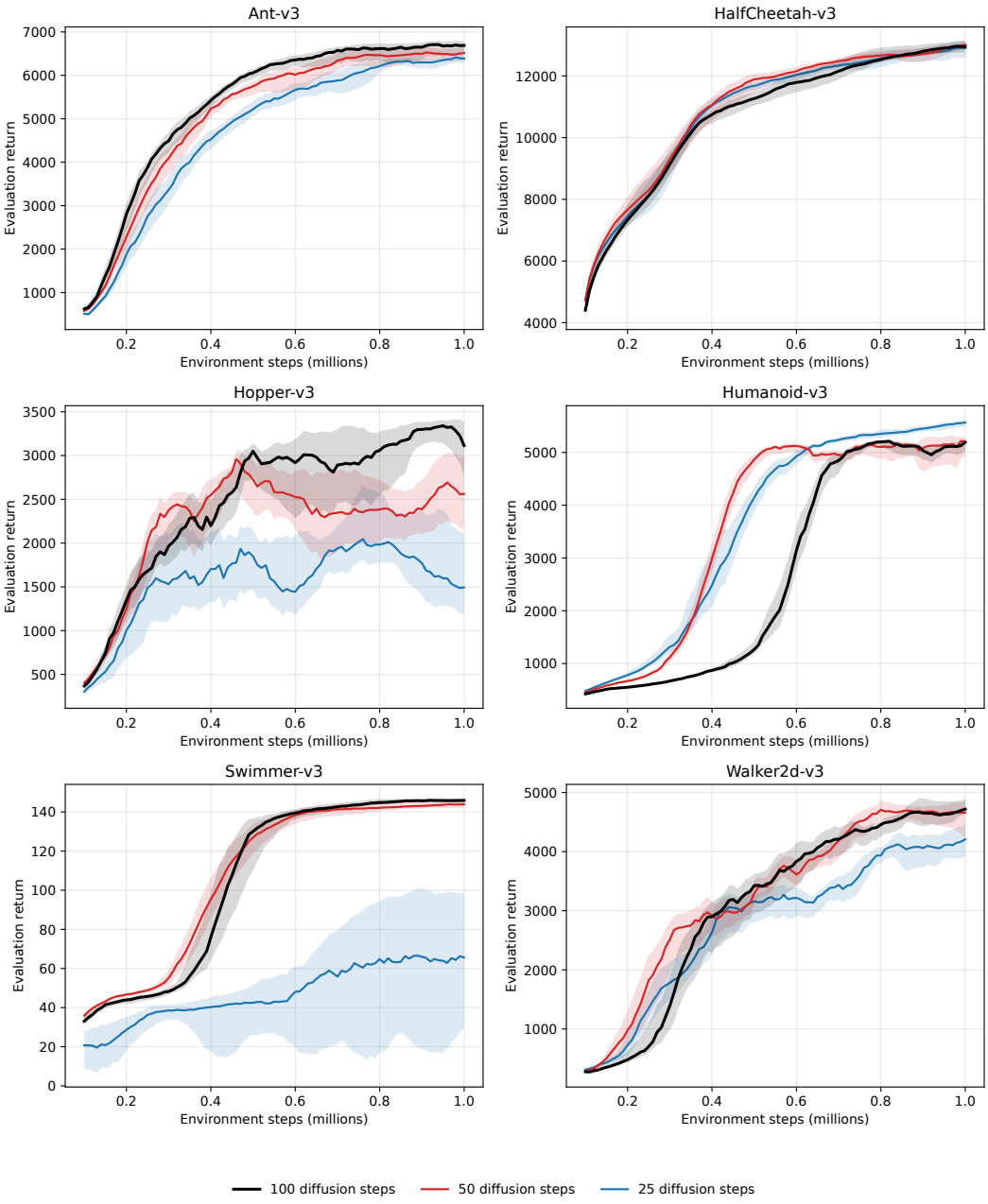


Figure 3: Comparison of MGMD performance with 100 vs 50 diffusion steps. Performance is mostly preserved for 5/6 environments when using 50, matching the baselines in terms of total number of score evaluations during each training step. At 25 diffusion steps, the performance noticeably drops for 3/6 environments.

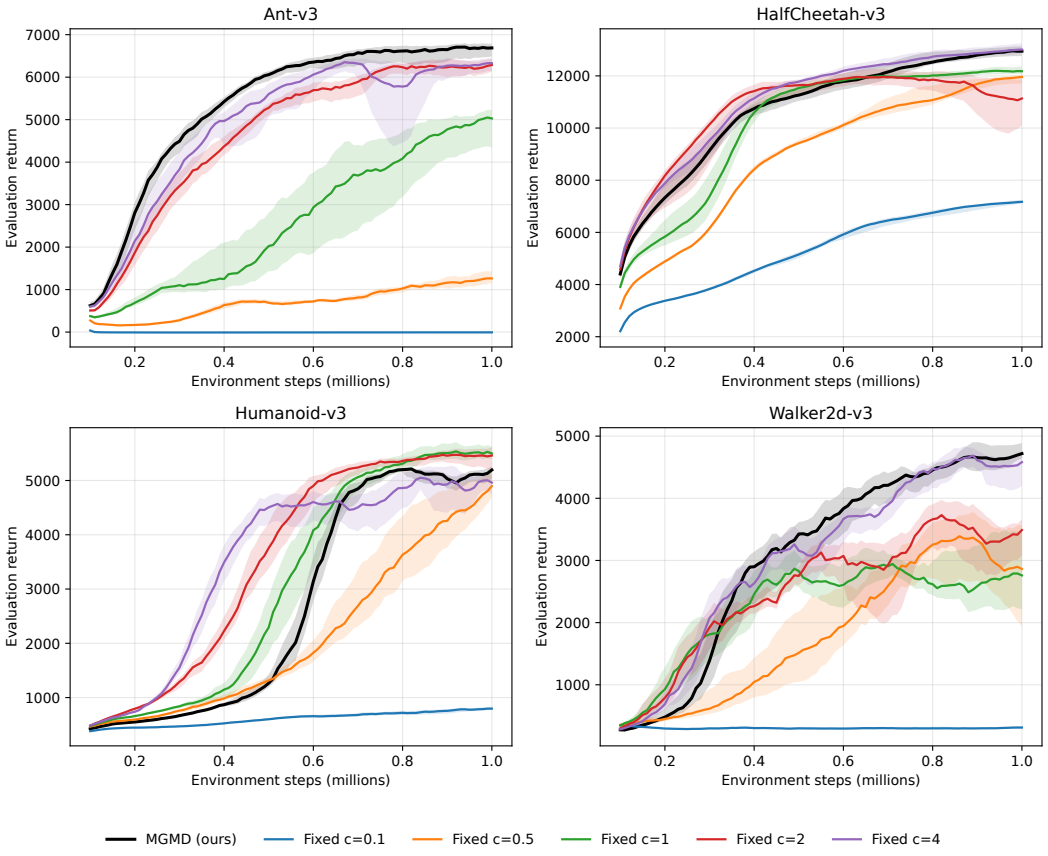


Figure 4: Ablation showing training curves for MALA step sizes fixed to $h_t = c(1 - \alpha_t)$ for various values of c. While fixed values of c perform well for individual environments, no value of c outperforms the adaptive step scheme uniformly on the tested environments.

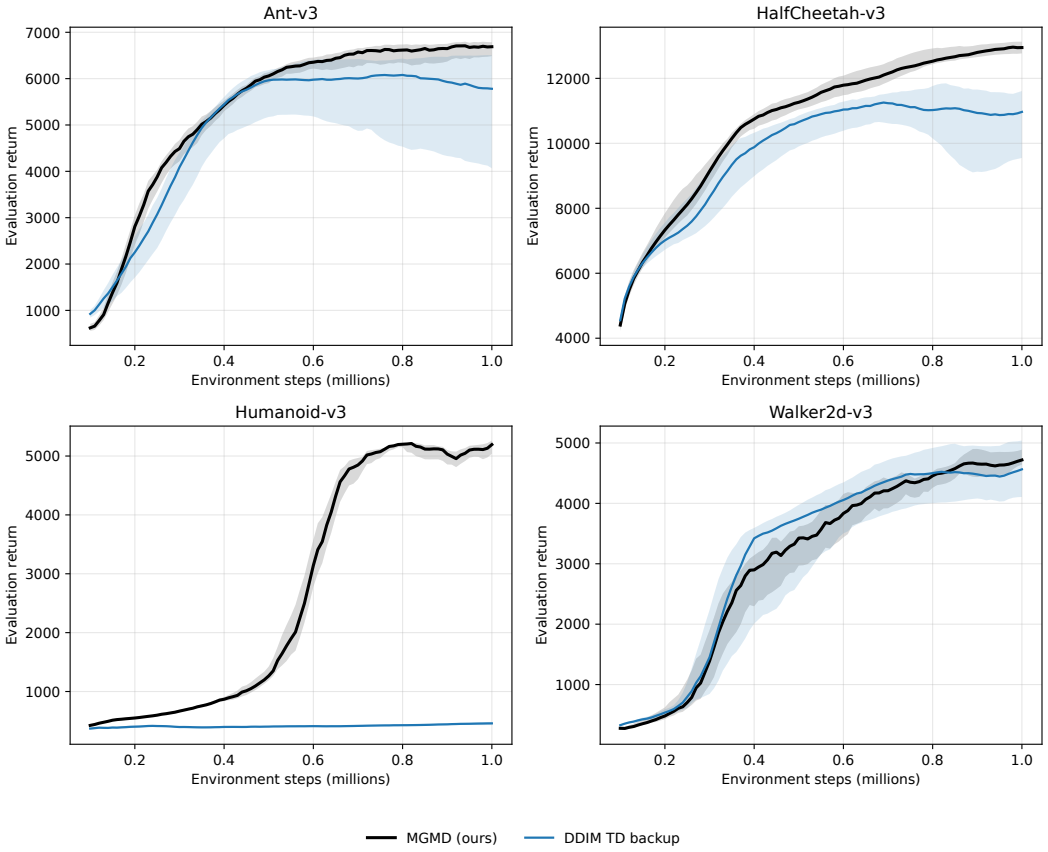


Figure 5: Ablation showing training curves when the Bellman targets for the critic are constructed using actions sampled from the base policy instead of the tilted policy. We notice that the performance degrades substantially on multiple environments, showing that training the critic on tilted actions is not just a computational convenience.